TCP/IP とデータ通信

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

Part No: 805–5857–10
1998 年 11 月

原典: *TCP/IP and Data Communications Administration Guide*

Part No: 805-4003-10

Revision A

Adobe PostScript

目次

はじめに

本書では、Solaris™ オペレーティング環境で TCP/IP プロトコル群を使用するネットワークの設定、保守、拡張の方法に関する、以下の内容について説明します。

- TCP/IP を扱う場合のネットワーク概念
- 新規にネットワークを設定するのに必要な作業
- ネットワークの保守
- ルーターを使ってインターネットワークを作ることによって、既存のネットワークを拡張する方法
- ポイントツーポイントプロトコル (PPP) を使ってリモートマシンがネットワークに接続できるようにする方法
- UNIX-to-UNIX Copy Program (UUCP) を用いたリモートマシンとの通信を設定する方法
- 動的ホスト構成プロトコル (DHCP) および、クライアントとサーバー

対象読者

本書には、広範な経験を持つネットワーク管理者向きの情報が記載されています。読者が Solaris 環境について十分に理解していること、ローカルマシンのほか、モデムなどの周辺デバイスの管理経験があることを前提としています。

新規のネットワークを設定しようとしているときは、まず本書をお読みになってから、Solaris 7 システム管理マニュアルセットに含まれる他のマニュアルに進んでく

ださい。既存のネットワークの管理や拡張を行おうとしている場合は、行いたい作業に該当する章をお読みください。

注 - Solaris やその他の UNIX ベースのネットワークを一度も使用したことのないサイトで、新規にネットワークを設定する必要がある場合は、Solaris ソフトウェアをインストールする前に、第 3 章をお読みください。この章には、『*Solaris* のインストール *(*上級編*)*』で説明されているインストール作業に関する重要な補足情報が記載されています。

各章を順番に読み進む必要はありませんが、どの章でも、それ以前の章の内容を理解しているものとして説明が進められています。

お読みになる前に

本書をお読みになる前に、以下のマニュアルの内容を理解しておいてください。

- 『*Solaris* のインストール *(*上級編*)*』
- 『*OpenWindows* ユーザーズガイド *(*上級編*)*』
- 『メールシステムの管理』
- 『*Solaris* のシステム管理 *(*第 *1* 巻*)*』
- 『*Solaris* のシステム管理 *(*第 *2* 巻*)*』

内容の紹介

本書は、以下の章で構成されています。

第 1 章では、ネットワーク管理者が行う一般的な作業について説明し、ネットワーキングの基本的な概念を紹介します。

第 2 章では、TCP/IP プロトコル群を形成するプロトコルを紹介します。

第 3 章では、新規のネットワークを設計するときに必要な考慮事項、たとえば、インターネットプロトコル (IP) のアドレス指定、ネットワークトポロジなどについて説明します。

第 4 章では、新規のネットワークにマシンを設定するための手順を示します。

第 5 章では、ルーターを用いてネットワークを拡張する方法について説明します。

第 6 章では、TCP/IP に関する問題の診断と修正のためのツールの使用方法について説明します。

第 7 章では、モデムと電話回線を使ってネットワークを拡張するために使用できる PPP データリンクプロトコルを紹介します。

第 8 章では、特定の PPP 構成を設計するときに必要な考慮事項について説明します。

第 9 章では、2 つの基本的な種類の PPP リンクを構成するための手順を示します。

第 10 章では、PPP に関する問題の診断と修正の方法について説明します。

第 11 章では、複雑な PPP リンクの設定について説明します。

第 12 章では、UUCP のデータベースファイルを設定する方法について説明します。

第 13 章では、UUCP の起動方法と、UUCP リンクに関する問題への対応方法について説明します。

第 14 章では、動的ホスト構成プロトコルを紹介します。このプロトコルによって、インターネットプロトコル (IP) アドレスと他のインターネット構成パラメータをホストが取得できます。その際、システム管理者が事前に構成を行う必要はありません。

第 15 章では、DHCP と初期のプロトコルの違いと、初期のプロトコルから DHCP への移行方法を説明します。

第 16 章では、DHCP が実行されるネットワークの設定、リース期間の方針の決定、BOOTP リレーエージェントの追加、DHCP で使用される一部のデータベース内でのマクロ作成の方法について説明します。

第 17 章では、DHCP の使用中に発生しうる問題を解決する方法を説明します。

付録 Aでは、Windows クライアントとして実行される PCNFSPro に固有の障害追跡手法について説明します。

関連マニュアル

ネットワークを設定後、Solaris オペレーティングシステムが提供するネットワークサービスを追加することができます。各種のサービスについては、システム管理マニュアルセットの中の下記のマニュアルに説明されています。

- 『*NFS* の管理』
- 『*Solaris* ネーミングの管理』
- 『*Solaris* ネーミングの設定と構成』
- 『*NIS+* への移行』

また、以下の書籍には、異種 TCP/IP ネットワークの管理に関するきわめて有用な情報が記載されています。

- Bart Anderson, Bryan Costales, Harry Henderson 著 *UNIX Communications* (Howard W. Sams & Company, 1987)
- William R. Cheswick and Steven M. Bellovin 著 *Firewalls and Internet Security* (Addison Wesley, 1994)
- Craig Hunt 著 *TCP/IP Network Administration* (O' Reilly & Associates, Inc., 1993)
- Ed Krol 著 *The Whole Internet User's Guide and Catalog* (O' Reilly & Associates, Inc., 1993)
- Tim O' Reilly and Grace Todino 著 *Managing UUCP and Usenet* (O' Reilly & Associates, Inc., 1992)
- W. Richard Stevens 著 *TCP/IP Illustrated, Volume 1, The Protocols* (Addison Wesley, 1994)

マニュアルの注文方法

SunDocs™ プログラムでは、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) の 250 冊以上のマニュアルを扱っています。このプログラムを利用して、マニュアルのセットまたは個々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、米国 SunExpress™, Inc. のインターネットホームページ `http://www.sun.com/sunexpress` にあるカタログセクションを参照してください。

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、または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は区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、または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越える場合、バックスラッシュは継続を示します。	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。上記の場合、*filename* は省略してもよいことを示します。

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

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

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

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が適宜、併記されています。
- 「x86」という用語は、一般に Intel 8086 ファミリに属するマイクロプロセッサを意味します。これには、Pentium、Pentium Pro の各プロセッサ、および AMD と 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。このマニュアルでは、このプラットフォームのアーキテクチャ全体を指すときに「x86」という用語を使用し、製品名では「Intel 版」という表記で統一しています。

パート I　TCP/IP ネットワークの設定と管理

パート I では、Solaris オペレーティング環境で TCP/IP を実行するネットワークの設定方法について説明します。ここでは、読者が UNIX に関する十分な知識を持ち、ローカル UNIX システムの管理についてある程度の経験を積んでいることを前提として、説明を進めます。ネットワーク管理の経験はなくてもかまいません。

第 1 章

ネットワーク管理の概要

この章では、ネットワーク管理者の役割を紹介します。新しくネットワーク管理者になった方にとっては、必要な作業の概略を理解するためにこの章の内容が役立ちます。またこの章には、本書を読むときに必要になるネットワークの基礎概念も示しています。すでに経験を積んだネットワーク管理者は、この章をとばして次の章に進んでもかまいません。

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 4 つの分野があります。

- 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 3 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 4 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
- ネットワークのセキュリティ
- 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

既存のネットワーク上に新しいホストを設定する方法については、第 4 章を参照してください。第 6 章には、ネットワーク上の問題を解決するためのヒントが記載されています。ネットワークサービスについての詳細は、『*NFS* の管理』、『*Solaris* ネーミングの管理』、『*NIS+* への移行』、『メールシステムの管理』を参照してください。セキュリティ関係の作業については、『*Solaris* のシステム管理 *(*第 *2* 巻*)*』を参照してください。

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っ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。

第 5 章は、インターネットを設定するための手順を説明しています。パート II には、可搬コンピュータ用のネットワーク接続の設定方法を説明しています。パート III は、UUCP を使って、ユーザーのマシンと他の UUCP システムとの間で情報を交換する方法について説明しています。

TCP/IP とは何か

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、イーサネット、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用いても、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 2 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 1–1 は、イーサネット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。

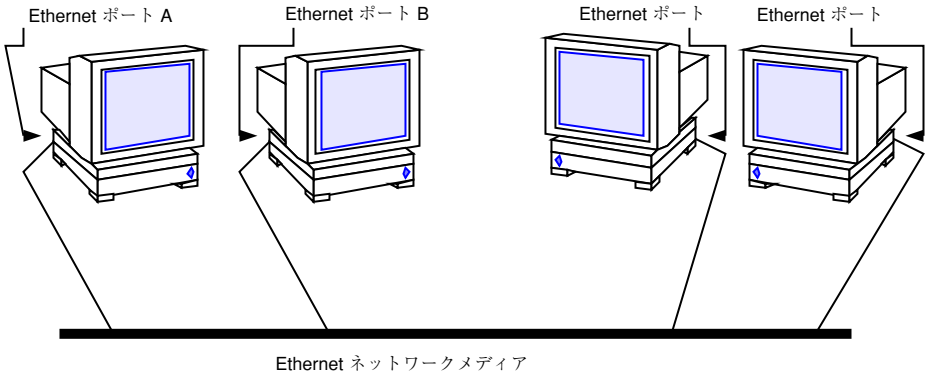


図 1–1　Solaris ローカルエリアネットワーク

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、イーサネット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イーサネット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。イーサネット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、イーサネット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 *2* 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

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 1–2 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。

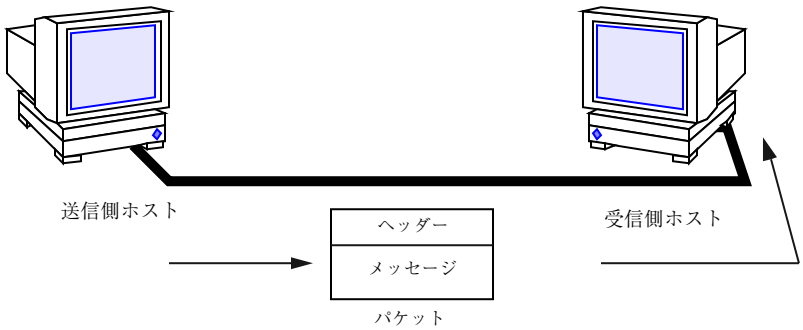


図 1–2　ネットワーク上での情報の転送

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：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」の同義語として使われます。TCP/IP の視点から見れば、ネットワーク上に存在する実体は、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用い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、第 4 章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、39ページの「IP 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「イーサネットアドレス」という言葉で表しています。イーサネットは、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わ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イーサネット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、最もよく使わ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 *2* 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パフォーマンス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 1 ギガ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(5)` 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)` 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。

```
example# ndd -set /dev/tcp tcp_host_param '129.154.0.0
mask 255.255.255.0 sendspace 8192 recvspace 8192'


example# ndd -set /dev/tcp tcp_host_param
'129.154.152.0 sendspace 32000 recvspace 32000'


example# ndd -set /dev/tcp tcp_host_param
'129.154.152.6 sendspace 128000 recvspace 128000 timestamp 1'
```

削除するには、次のように指定します。

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

example# ndd -set /dev/tcp tcp_host_param '129.154.152.0 delete'

example# 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 選択式応答のサポート

TCP 選択式応答 (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)` 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 2 章

TCP/IP プロトコル群

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 1 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、この章をとばして行いたい作業に該当する章に進んでもかまいません。

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われています。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われています。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 3 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 2–1 に示すように、上 (層 7) から下 (層 1) へ並べて表します。

表 2–1　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、またまったく使わ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 2–2 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 2–2 TCP/IP プロトコルスタック

OSI 参照の層番号	対応する **OSI** 層	**TCP/IP** 層	**TCP/IP** プロトコルの例
5,6,7	アプリケーション、セッション、プレゼンテーション	アプリケーション	NFS、NIS+、DNS、`telnet`、`ftp`、“r”(リモート)コマンド[1]、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	イーサネット (IEEE 802.3) トークンリング、RS-232、その他

1.“r” コマンドには、`rlogin`、`rsh`、`rcp` があります。

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、イーサネット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、イーサネット 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)、ARP プロトコル、ICMP プロトコル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 (IP アドレス指定については、第 3 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、28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。

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、イーサネットアドレス (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)

第 6 章に、エラー検出のために ICMP を使用するオペレーティングシステムコマンドに関する詳細が記載されています。

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なおします。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 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 を使っ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、『*OpenWindows* ユーザーズガイド *(*上級編*)*』および、`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 アドレスとイーサネット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、『*NFS* の管理』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、第 5 章で説明します。

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 2–1 に、この対話がどのように行われるかを示します。

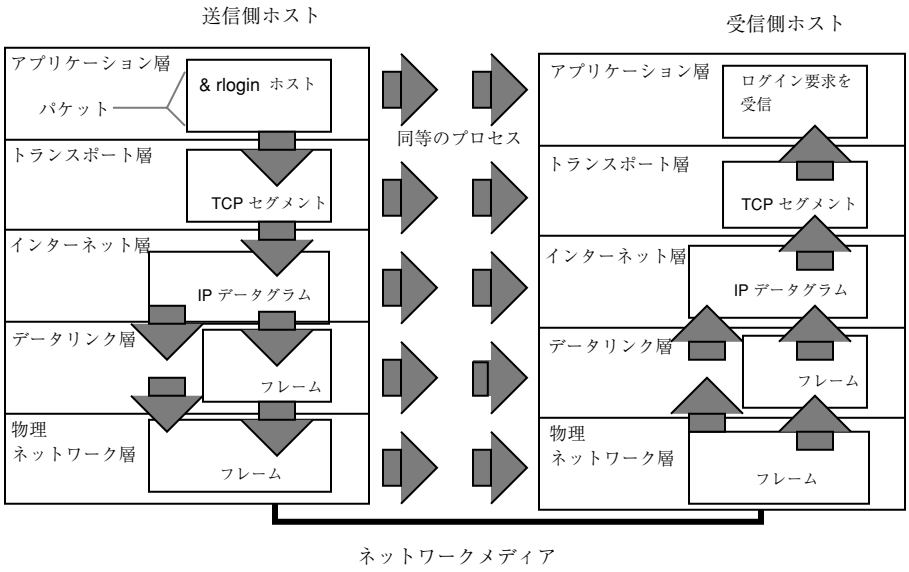


図 2–1　TCP/IP スタックを通過するパケット

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 2–1 に示すように層によって異なります。

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

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

トランスポート層 — データのカプセル化の開始

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

TCP のセグメンテーション

TCP はしばしば「接続指向型」プロトコルと呼ばれますが、これは、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 2–1 に、TCP プロトコルが `rlogin` コマンドからのストリームをどのように受け取るかを示してあります。TCP は、アプリケーション層から受け取ったデータをセグメントに分割し、各セグメントにヘッダーを添付します。

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

TCP 接続の確立

TCP は、受信側ホストでデータ受信の準備が整っているかどうかを判別するためにも、セグメントを使用します。送信側 TCP は、接続を確立するために、受信側ホスト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 2–1 に示したように、TCP と UDP はどちらもセグメントとパケットを下位のインターネット層に送り、セグメントとパケットはそこで IP プロトコルにより処理されます。IP は、セグメントとパケットを IP データグラムと呼ばれる単位に形式化して、配送の準備を整えます。次に、IP はデータグラムの IP アドレスを判別して、受信側ホストへの効率的な配送ができるようにします。

IP データグラム

IP は、TCP または UDP が付加した情報に付け加える形で、セグメントまたはパケットのヘッダーに「IP ヘッダー」を付加します。IP ヘッダーには、送信側ホストと受信側ホストの IP アドレス、データグラムの長さ、データグラムのシーケンス番号が含まれます。これらの情報が付加されるのは、データグラムがネットワーク

パケットとしての許容バイトサイズを超過してフラグメント化が必要になった場合に備えるためです。

データリンク層 — フレーミングの実施

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

物理ネットワーク層 — フレームの転送準備

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

受信側ホストでのパケットの取り扱い

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

1. 物理ネットワーク層はフレーム形式のパケットを受け取ります。パケットの CRC を計算し、データリンク層にフレームを送ります。
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 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。本書で説明されていない特別な情報は、以下に挙げる情報源から入手できると考えられます。

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*』 - この書籍には、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*』 - この書籍では、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*』 - この書籍は、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 7 システム管理者セットに含まれる他のマニュアルでも、随所で関連の RFC が参照されています。

RFC の入手方法

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*`.txt`

  `end`

  または

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

  `end`

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

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 3 章

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項について説明します。これらの事項を解決後、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。

まだ TCP/IP の基本事項に慣れていない方は、第 2 章を参照してください。

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、イーサネット、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、ディスクレス、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。

以下第 3 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。

IP アドレス番号の構成部分

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。

IP 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。IP 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 IP アドレスの 1 バイトを表します。このような形式で IP 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。

IP 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 3–1 に、129.144.50.56 という典型的な IP アドレスの構成部分を示します。

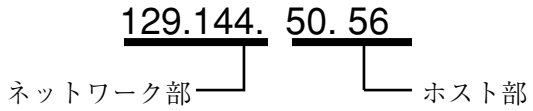


図 3–1　IP アドレス番号の構成部分

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 3–1 では、ネットワーク部は IP アドレスの 2 バイトを占めています。

ホスト部

IP 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。IP 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、IP アドレス空間の有効率を最大限にすることができます。ネットワーク識別子として使用した場合、アドレスの指定した部分がサブネット番号になります。サブネット番号は、ネットマスクを使って作成します。ネットマスクは、IP 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、63ページの「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 IP 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、IP 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、下の 図 3–2 に示すように、IP アドレスのホスト部です。

クラス A アドレス

図 3–2　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 IP 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 3–3 に、クラス B のアドレスを示します。

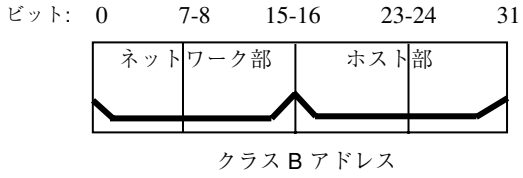


図 3–3　クラス B アドレスのバイト割り当て

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、IP 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 3–4 に、クラス C アドレスのバイトを示します。

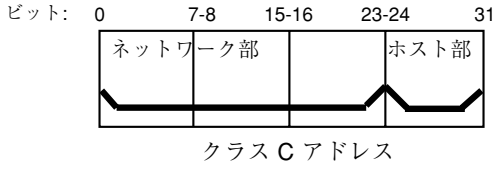


図 3–4　クラス C アドレスのバイト割り当て

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。

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。

IP 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、IP アドレスのホスト部をどのように割り当てるかについて、計画を立てることができます。

表 3–1 は、IP 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、IP 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 3–1 IP 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*

IP 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 3–2 は、IP 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 3–2　使用できる番号の範囲

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

ネットワークインタフェースへの IP アドレスの適用法

7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、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。第 5 章で説明したように、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、よく使われる 2 つのイーサネットインタフェースのデバイス名です。

注 - 本書では、イーサネット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。

ネットワーク上の実体への名前付け

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 を用い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実体の集合です。ホストは名前が付けられた 1 個の実体です。ルーターも名前が付けられた 1 個の実体です。さらに、ネットワークも名前が付けられた 1 個の実体です。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実体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、イーサネット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` データベースと呼ばれるネットワークデータベースにこの情報を入れます。`hosts` データベースは、ネットワーク上での TCP/IP の動作に必要な情報が含まれている一組のネットワークデータベースの 1 つです。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、第 4 章に詳しい説明があります。

ネームサービスとしての NIS、NIS+、DNS の使用

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」の概念に関する詳しい説明も出ています。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。

ネームサービスとしてのローカルファイルの使用

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っ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、63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 インタフェースを用いた匿名 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

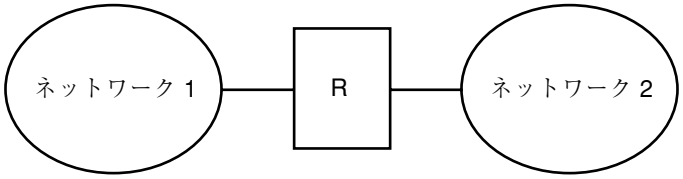
TCP/IP から見た場合、ネットワーク上に存在するのは、2 つの種類の実体、つまりホストとルーターだけです。ホストはすべてのネットワークに必要がありますが、ルーターはすべてのネットワークに必要なわけではありません。ルーターを使用するかどうかは、ネットワークの物理的なトポロジによって異なります。この節では、ネットワークトポロジとルーティングの概念を紹介します。この概念は、既存のネットワークに別のネットワークを追加しようとするときに、重要な意味を持ちます。

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実体です。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、第 5 章の説明に従って、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 3–5 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 3–5　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、図 3–6 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 の間の送信経路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。

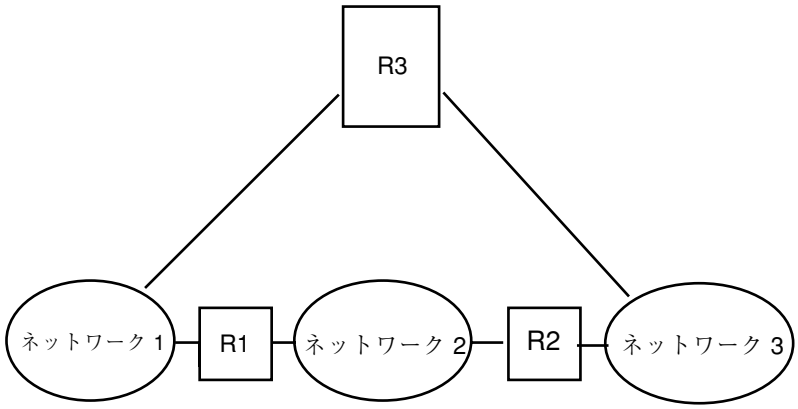


図 3–6　ネットワーク間のパスの追加

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、 第 5 章を参照してください。

図 3–7 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。

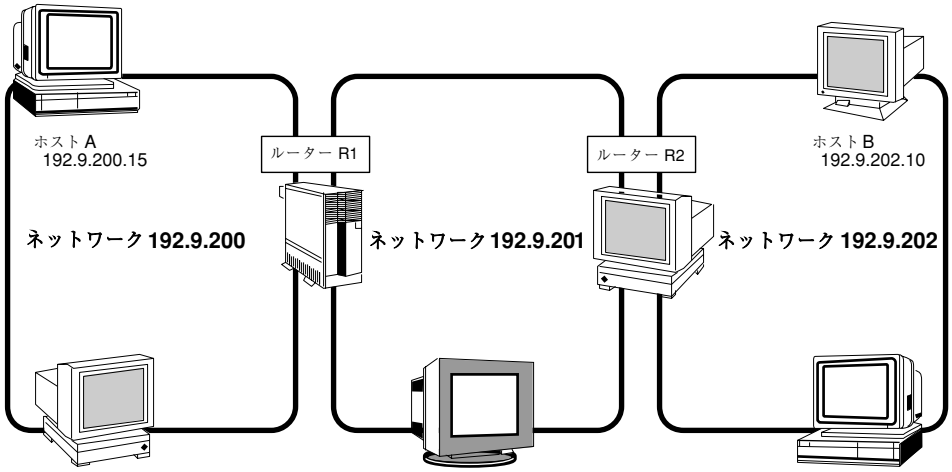


図 3–7　相互接続された 3 つのネットワーク

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 IP 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 IP 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「次の中継」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 4 章

ネットワーク上での TCP/IP の構成

ネットワーク管理の第 2 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 の構成方法を次の事項に分けて説明します。

- ネットワーク上の各マシンについてのホスト構成モードの決定
- ネットワークのサブネットマスクの設定 (オプション)
- ローカルファイルモードで実行されるマシン上での TCP/IP の構成
- ネットワーク構成サーバーの構成
- ネットワーククライアントモードで実行されるマシン上での TCP/IP の構成
- ネットワーク用に選択したネームサービスに基づくネットワークデータベースの編集
- ネームサービススイッチファイルの構成
- 52ページの「TCP/IP の構成の前に」
- 53ページの「ホスト構成モードの決定」
- 57ページの「TCP/IP 構成ファイル」
- 59ページの「hosts データベース」
- 63ページの「ネットワークマスクの作成」
- 67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」
- 69ページの「nsswitch.conf ファイル — 使用するネームサービスの指定」
- 76ページの「ネットワーク構成手順」
- 83ページの「標準 TCP/IP サービスの構成」

TCP/IP の構成の前に

TCP/IP ソフトウェアを構成する前に、以下のことをしておく必要があります。

1. 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します。(詳細は、46ページの「ネットワークトポロジ」を参照してください)。
2. 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します。(36ページの「ネットワーククラス」を参照してください)。
3. 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します。(ハードウェアのマニュアルと、 46ページの「ネットワークトポロジ」を参照してください)。
4. 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、それを実行します。(ルーターについては、 第 3 章と第 5 章を参照。購入したネットワークインタフェースをマシンにインストールしてある場合は、ソフトウェア構成要件についてそのインタフェースのマニュアルを参照してください)。
5. 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ます。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ます。(39ページの「IP アドレス指定スキーマの設計」を参照してください)。
6. 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ます。(39ページの「IP アドレス指定スキーマの設計」を参照してください)。
7. 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します。(『*Solaris* ネーミングの管理』を参照してください)。
8. 必要なら、ネットワークで使用するドメイン名を選択します。(『*Solaris* ネーミングの管理』を参照してください)。
9. 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します。『*Solaris* のインストール *(*上級編*)*』を参照してください。

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

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、57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のですが、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、既知のイーサネットアドレス (48 ビット) を IP 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` デーモンを使って使用可能にできます。詳細は、`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、ディスクレスクライアントが必要とするブート用パラメータを提供します。このサービスを実施するのは `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ィスクレスマシンや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。

ディスクレスブート

システムをディスクレスブートに設定する方法は、『*Solaris* のシステム管理 *(*第 *1* 巻*)*』で説明しています。

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 4–1 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。このマシンは、ディスクレスクライアント `ahaggar` にサービスを提供します。`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 4–1 に示すネットワークにもとづいて説明しています。

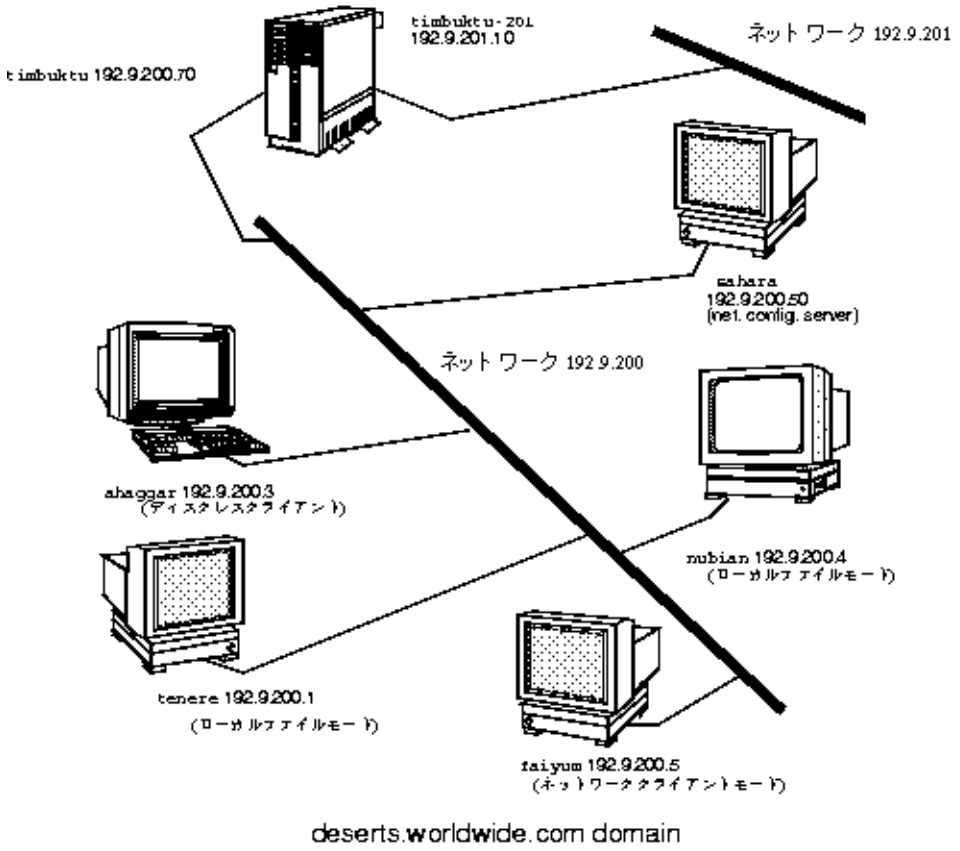


図 4–1　サンプルネットワーク内のホスト

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
- 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、「TCP/IP 構成ファイル」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、67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。

/etc/hostname.*interface* ファイル

このファイルは、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。

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 IP アドレスのどちらかです。たとえば、ahaggar というマシンの一次ネットワークインタフェースが smc0 であるとすると、/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 ahaggar 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がある場合

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って作成します。Solaris インストールプログラムは、追加のファイルは作成しません。

たとえば、図 4–1 に示したマシン timbuktu について考えてみましょう。このマシンは 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 IP 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/nodename ファイル

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

/etc/defaultdomain ファイル

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

図 4–1 では、ネットワークはドメイン deserts.worldwide に属しており、このドメインは .com ドメインとして分類されています。したがって、/etc/defaultdomain には deserts.worldwide.com というエントリが入ります。ネットワークドメインについての詳細は、『*Solaris* 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 4–1 で、ネットワークインタフェース `le1` は、マシン `timbuktu` をネットワーク 192.9.201 に接続しています。このインタフェースには、`timbuktu-201` という一意な名前が付いています。したがって、ネットワーク 192.9.201 にあってローカルファイルモードで構成されているマシンについては、`/etc/defaultrouter` に `timbuktu-201` という名前が エントリとして入ります。

hosts データベース

`hosts` データベースには、ネットワーク上のマシンの IP 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 IP 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)。

IP-address hostname [nicknames] [#comment]

IP-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 IP 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、コメントを入れることができます (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、IP アドレス、ホスト名) が入っています。

たとえば、図 4–1 に示したマシン ahaggar 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。

コード例 **4–1**　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   ahaggar                      #host name
```

ループバックアドレス

コード例 4–1 では、IP 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。98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

IP アドレス 192.9.200.3 と名前 ahaggar 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 IP 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、第 5 章を参照してください)。

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

コード例 4–2 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ってネットワーク 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 IP 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。

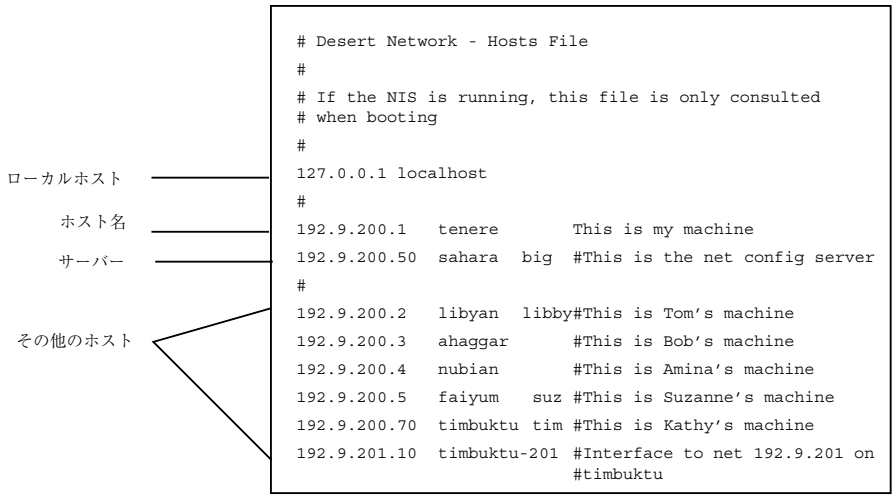
- ループバックアドレス
- ローカルマシン (一次ネットワークインタフェース) の IP アドレスとホスト名

- 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 IP 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 IP 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 IP アドレスとホスト名
- このマシンでホスト名を使って参照したいマシンの IP アドレス

コード例 4–3 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 IP 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 IP 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 IP アドレスを入手します。

コード例 4–3 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




```
# Desert Network - Hosts File
#
# If the NIS is running, this file is only consulted
# when booting
#
127.0.0.1 localhost
#
192.9.200.1   tenere        This is my machine
192.9.200.50  sahara  big  #This is the net config server
#
192.9.200.2   libyan  libby#This is Tom's machine
192.9.200.3   ahaggar       #This is Bob's machine
192.9.200.4   nubian        #This is Amina's machine
192.9.200.5   faiyum   suz #This is Suzanne's machine
192.9.200.70  timbuktu tim #This is Kathy's machine
192.9.201.10  timbuktu-201  #Interface to net 192.9.201 on
                            #timbuktu
```

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 IP 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。

IP アドレス空間を有効活用できることのほかに、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って、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。

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 IP 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。その後で、66ページの「ネットワークにサブネットを追加する方法」に示す手順を実施します。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。

35ページの「IP アドレス番号の構成部分」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 IP 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 IP 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って IP 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 進数は計算機を使って換算できます。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 IP アドレス 129.144.41.101 に適用した場合、結果の IP 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。

11111111.11111111.11111111.00000000 (ネットマスク)

AND

10000001.10010000.00101001.01100101 (IP アドレス)

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。IP 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 ファイルの編集

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。

コード例 4–4 に示すのは、クラス B ネットワークの場合のサンプルの `/etc/inet/netmasks` ファイルです。

コード例 4–4 クラス B ネットワークの場合の `/etc/inet/netmasks` ファイル

```
## The netmasks file associates Internet Protocol (IP) address
# masks with IP network numbers.
#
#  network-number netmask
```

```
#
# Both the network-number and the netmasks are specified in
# ''decimal dot'' notation, e.g:
#
#        128.32.0.0   255.255.255.0
         129.144.0.0  255.255.255.0
```

このファイルが存在しない場合は、次の構文を使っ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。

▼ ネットワークにサブネットを追加する方法

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には、次の手順を使用します。

1. 新しいサブネットトポロジについて決定します。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ます。
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ます。
3. 手作業で **TCP/IP** を構成している場合は、/etc/inet/netmasks ファイルを修正します。そうでない場合は、**Solaris** インストールプログラムにネットマスクを与えます。

4. すべてのホストで、新しいホストアドレスを反映するように /etc/inet/hosts ファイルを修正します。

5. すべてのマシンをリブートします。

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
- 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 IP 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 IP 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 4–2 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。

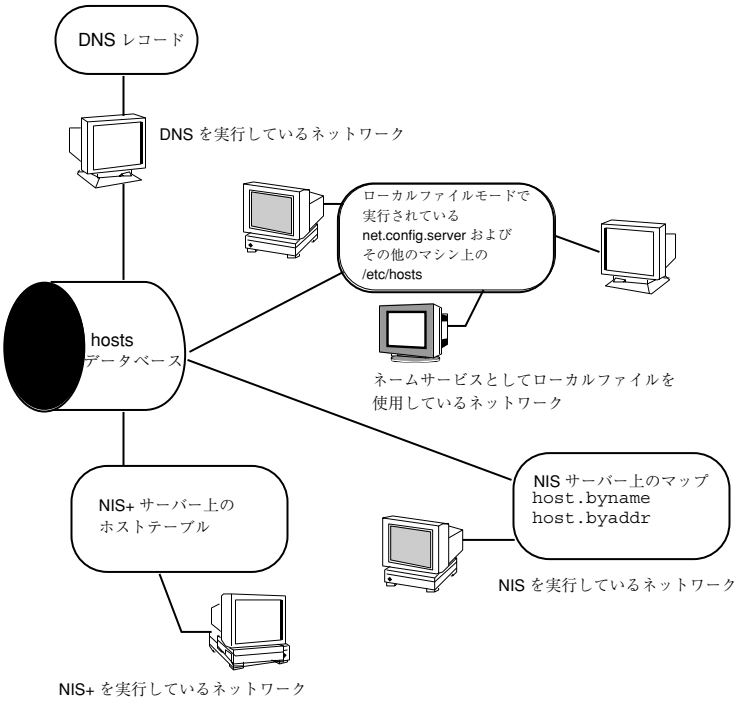


図 4–2　ネームサービスが使用する hosts データベースの形式

表 4–1 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 4–1 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.ord_dir	hosts.byaddr hosts.byname
netmasks	/etc/inet/netmasks	netmasks.ord_dir	netmasks.byaddr
ethers	/etc/ethers	ethers.ord_dir	ethers.byname ethers.byaddr
bootparams	/etc/bootparams	bootparams.ord_dir	bootparams
protocols	/etc/inet/protocols	protocols.ord_dir	protocols.byname protocols.bynumber
services	/etc/inet/services	services.ord_dir	services.byname
networks	/etc/inet/networks	networks.ord_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 データベースに関する情報は、59ページの「hosts データベース」を、netmasks データベースに関する情報は、62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 ファイルはコード例 4–5 のようになります。

コード例 4–5 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用の nsswitch.conf ファイル

```
# /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。

コード例 4–5 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、ディスクレス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。コード例 4–6 に、正しい検索順序を示します。

コード例 4–6 NIS を実行するネットワークでのディスクレスクライアント用の nsswitch.conf

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

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

automount:      files nis
aliases:        files nis

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

```
sendmailvars:  files
```

ネームサービススイッチについての詳細は、『*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ディスクレスのクライアント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、80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。

コード例 4–7　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ディスクレスホストに指示します。

bootparams のワイルドカードエントリ

ディスクレス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、ホスト名をイーサネット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またはディスクレスマシンを構成する場合だけです。

RARP は、このファイルを使って、イーサネットアドレスを 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、ホストのイーサネットアドレスです。

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。

イーサネット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イーサネット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 データベース内の一次名に一致していることを確かめてください (コード例 4–8)。

コード例 **4–8**　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) 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って状態テーブルを作成します。

コード例 4–9 に /etc/networks ファイルのサンプルを示します。

コード例 **4–9**　/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) マニュアルページに記載されています。コード例 4–10 に、/etc/inet/protocols ファイルの例を示します。

コード例 4–10　/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。 コード例 4–11 に、典型的な /etc/inet/services ファイルからの抜粋を示します。

コード例 **4–11**　/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
```

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、単に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だけです。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、第 5 章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。

▼ ローカルファイルモードの場合のホストの構成方法

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

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

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

   (一次ネットワークインタフェースについては、Solaris インストールプログラムが自動的にこのファイルを作成します)。

   詳細は、57ページの「`/etc/hostname.`*interface* 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、59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。

---

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、58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、59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。81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

いくつかのホストをネットワーククライアントとして構成することを予定している場合は、ネットワーク上のマシンの少なくとも 1 つは、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 (方法については、54ページの「ネットワーク構成サーバー」を参照してください)。

ネットワーク構成サーバーの設定には、次のような操作が必要です。

1. ネットワーク構成デーモンが動作するようにします。
   - in.tftpd
   - in.rarpd
   - rpc.bootparamd
2. 構成サーバー上のネットワーク構成ファイルを編集し保守します。

   79ページの「ネットワーク構成サーバーの設定方法」では、ネットワーク構成サーバーをすでにローカルファイルモード用として設定してあるものとします。

▼ 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。

   72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。

ディスクレスクライアント、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* 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。76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。

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。

   (60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。

   ただし、ディスクレスクライアント (NFS マウントされたルートファイルシステムを持つマシン) の場合は、クライアントのルートファイルシステム (ほとんどの場合ネットワーク構成サーバ) を提供するサーバーの名前と IP アドレスを入力します。

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、それを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。

ネットワーククライアントのインストール後に

各ネットワーククライアントマシン上のファイルを編集し終わったら、ネットワーク構成サーバーで次の作業を行います。

1. ethers データベースと hosts データベースにそのホストのエントリを追加します。

2. bootparams データベースにそのホストのエントリを追加します。

   操作を簡単にするために、bootparams データベースには、各ホストのエントリを個別に入力する代わりに、ワイルドカードを入力することができます。その例については、72ページの「bootparams データベース」を参照してください。

3. サーバーをリブートします。

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 フラグを使って構成できます。

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。ログ記録を行うには、実行中の inetd を終了し、次のように入力します。

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

t スイッチは、inetd に TCP 接続トレースを開始させます。

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。

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。

第 5 章

ルーターの構成

この章では、ルーティングプロトコルについて説明し、特に TCP/IP ネットワーク上にルーターを構成するための手順を示します。ルーターとは、複数のネットワークインタフェースを持ち、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送信するマシンです。最も一般的なルーターの種類としては、カードスロットに増設ネットワークインタフェースを持つコンピュータと、各種のメーカーから販売されている専用ルーターの 2 種類があります。

この章では、ルーティングの原理については説明していません。ルーティングの原理については、46ページの「ネットワークトポロジ」で説明されています。また、48ページの「ルーターがどのようにパケットを転送するか」には、ルーティングに関する基本事項の説明があります。サブネットの作成方法については、62ページの「`netmasks` データベース」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送信経路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「スペース節約モード」) 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送信経路を 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送信経路に入ります。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。

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、7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、77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

ルーターとして構成したいマシンでスーパーユーザーになり、次のようにします。

1. インストールされている各ネットワークインタフェースについて、`/etc/hostname.`***interface*** ファイルを作成します。

   たとえば、`hostname.ie0` と `hostname.ie1` を作成します (詳細は、57ページの「`/etc/hostname.`*interface* ファイル」を参照してください)。

2. 対応するインタフェース用として選択したホスト名を各ファイルに入力します。

   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。

3. 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。

4. 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。

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。89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。

ルーティングプロトコルの自動選択

起動スクリプトは、次に、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、それとも静的ルーティングを使用するかを決める必要があります。

ホストで静的ルーティングを選択するには

ホストがディスクレスクライアントかネットワーククライアントである場合は、単に、ネットワーク上のルーターを `/etc/defaultrouter` に追加するだけですみます (59ページの「`/etc/defaultrouter` ファイル」を参照してください)。すると、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。

ホストで動的ルーティングを選択するには

ディスクレスクライアントまたは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には、`/etc/defaultrouter` ファイルが空であることが必要です。使用する動的ルーティングの種類は、次の基準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。そのためには、`/etc/gateways` という名前のファイルを作成し、それを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、137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

マルチホームホストにしたいマシンでスーパーユーザーになり、次のことを行います。

1. 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。

2. 次のようにタイプします。

   % touch /etc/notrouter

   これで、`/etc/notrouter` と呼ばれる空のファイルが作成されます。

3. 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。

スペース節約モードをオンにする方法

スペース節約モードでは、デフォルトの送信経路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、スペース節約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、スペース節約モードをオンにします。そのためには、/etc/rc2.d/S69inet 起動スクリプトの中の次の行を編集します。

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

これを次のように変更します。

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

ホストでの ICMP Router Discovery を止める方法

ルーターの信頼性に関連した理由により、ホストに RDISC を使用させたくない場合があります。RDISC を止めるには、ホストの /usr/sbin/in.rdisc の名前を、何か別の名前、たとえば /usr/sbin/in.rdisc.saved に変更し、ホストをリブートします。

ルーターでの ICMP Router Discovery を止める方法

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ほかにないときに、Solaris ルーターを 1 つインストールすると、デフォルトにより、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。そのためには、そのルーターの /usr/bin/in.rdisc ファイルの名前を別の名前に変更し、ルーターをリブートします。

第 6 章

TCP/IP の障害追跡

この章では、TCP/IP ネットワークの一般的な障害追跡の方法と、そのために使用できるツール (ping、ifconfig、netstat、route など) について説明します。

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パフォーマンス低下の原因です。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 コマンドを使ってネットワーク情報を表示します。
2. hosts データベースを検査して、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。
3. RARP を実行している場合は、 ethers データベース内のイーサネット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、22ページの「ICMP プロトコル」を参照してください)。

次のように入力したとします。

```
$ ping elvis
```

ホスト elvis が動作していれば、次のメッセージが表示されます。

```
elvis is alive
```

これは、elvis が ICMP の要求に応答したことを示します。しかし、elvis 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 elvis
```

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 の s オプションを使用して、問題を追求することができます。たとえば次のように入力します。

```
$ ping -s elvis
```

ping は、ユーザーが割り込み文字を送るかまたはタイムアウトが生じるまで、elvis 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、次の 98ページの「ifconfig コマンド」と、99ページの「netstat コマンド」を参照してくだい。

ifconfig コマンド

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

ifconfig *interface-name [protocol_family]*

特定のインタフェース、たとえば le0 に関する情報を表示したい場合は、次のように入力します。

```
$ ifconfig le0
```

le0 インタフェースの場合は、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速度が 1500 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 (この場合はイーサネット) です。

ifconfig の便利なオプションの 1 つに –a 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。たとえば、**ifconfig -a** と入力したとします。

```
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
```

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。

実行され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われるオプションは、 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。

プロトコル別統計の表示

netstat -s オプションは、UDP、TCP、ICMP、IP プロトコルについて、プロトコル別の統計を表示します。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
      tcpCurrEstab        =      1      tcpOutDataBytes     = 10547
      tcpOutDataSegs      =    288      tcpRetransBytes     =  8376
      tcpRetransSegs      =     29      tcpOutAckDelayed    =    23
      tcpOutAck           =     27      tcpOutWinUpdate     =     2
      tcpOutUrg           =      2      tcpOutControl       =     8
      tcpOutWinProbe      =      0      tcpOutFastRetrans   =     1
      tcpOutRsts          =      0
      tcpInSegs           =    563      tcpInAckBytes       = 10549
      tcpInAckSegs        =    289      tcpInAckUnsent      =     0
      tcpInDupAck         =     27      tcpInInorderBytes   =   673
      tcpInInorderSegs    =    254      tcpInInorderBytes   =   673
      tcpInUnorderSegs    =      0      tcpInUnorderBytes   =     0
      tcpInDupSegs        =      0      tcpInDupBytes       =     0
      tcpInPartDupSegs    =      0      tcpInPartDupBytes   =     0
      tcpInPastWinSegs    =      0      tcpInPastWinBytes   =     0
      tcpInWinProbe       =      0      tcpInWinUpdate      =   237
      tcpInClosed         =      0      tcpRttNoUpdate      =    21
      tcpRttUpdate        =    266      tcpTimRetrans       =    26
      tcpTimRetransDrop   =      0      tcpTimKeepalive     =     0
      tcpTimKeepaliveProbe=      0      tcpTimKeepaliveDrop =     0

IP
```

```
      ipForwarding        =     2      ipDefaultTTL        =   255
      ipInReceives        =  4518      ipInHdrErrors       =     0
      ipInAddrErrors      =     0      ipInCksumErrs       =     0
      ipForwDatagrams     =     0      ipForwProhibits     =     0
      ipInUnknownProtos   =     0      ipInDiscards        =     0
      ipInDelivers        =  4486      ipOutRequests       =  2805
      ipOutDiscards       =     5      ipOutNoRoutes       =     0
      ipReasmTimeout      =    60      ipReasmReqds        =     2
      ipReasmOKs          =     2      ipReasmReqds        =     2
      ipReasmDuplicates   =     0      ipReasmFails        =     0
      ipFragOKs           =    20      ipReasmPartDups     =     0
      ipFragCreates       =   116      ipFragFails         =     0
      tcpInErrs           =     0      ipRoutingDiscards   =     0
      udpInCksumErrs      =     0      udpNoPorts          =    33
      rawipInOverflows    =     0      udpInOverflows      =     6

ICMP

      icmpInMsgs          =     0      icmpInErrors        =     0
      icmpInCksumErrs     =     0      icmpInUnknowns      =     0
      icmpInDestUnreachs  =     0      icmpInTimeExcds     =     0
      icmpInParmProbs     =     0      icmpInSrcQuenchs    =     0
      icmpInRedirects     =     0      icmpInBadRedirects  =     0
      icmpInEchos         =     0      icmpInEchoReps      =     0
      icmpInTimestamps    =     0      icmpInTimestampReps =     0
      icmpInAddrMasks     =     0      icmpInAddrMaskReps  =     0
      icmpInFragNeeded    =     0      icmpOutMsgs         =     7
      icmpOutDestUnreachs =     1      icmpOutErrors       =     0
      icmpOutDrops        =     5      icmpOutTimeExcds    =     0
      icmpOutParmProbs    =     0      icmpOutSrcQuenchs   =     6
      icmpOutRedirects    =     0      icmpOutEchos        =     0
      icmpOutEchoReps     =     0      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
```

ネットワークインタフェースの状態の表示

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示します。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

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。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 フラグは送信経路が up 状態であること、G フラグは送信経路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 送信経路当たりの有効ユーザーの数、Use 列は 1 送信経路当たりの送信パケット数を示します。最後の Interface 列は、送信経路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、すべてのパケット転送も含めてそのデーモンの動作をログに記録することができます。ルーティングデーモンの動作のログファイルを作成するには、routed デーモンを起動するときにファイル名を指定します。たとえば次のように入力します。

```
# /usr/sbin/in.routed /var/routerlog
```

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。

▼ システムから全パケットを確認する方法

1. netstat -i と入力し、システムに接続されたインタフェースを検索します。

   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。

2. **root** になって snoop と入力します。

   Ctrl -C で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 129.146.82.36, npmpk17a-82 ?
```

3. 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 129.146.82.36? 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. **root** になって snoop -o ***filename*** の形式で入力します。たとえば、次のように入力します。

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

(続く)

```
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)。

2. snoop -i ***filename*** の形式で入力し、ファイルを検査します。

```
# 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. **root** になって snoop をオプション付きで実行し、結果をファイルに保存します。

3. 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の詳細は、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 にしてプローブの送信を開始し、プローブ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

以下のコマンドを入力するのが、traceroute ユーティリティを実行する最も簡単な方法です。

```
traceroute hostname
```

上記の *hostname* は、宛先ホストの名前を示します。

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
```

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

パートII　PPP によるネットワークの拡張

パート II では、ネットワーク上で非同期 PPP 通信リンクを設定し管理する方法について説明します。この章の説明は、読者が、TCP/IP に関する十分な知識を持つ熟練したネットワーク管理者であることを前提としています。

第 7 章

PPP の概要

この章では、TCP/IP プロトコル群に含まれているデータリンクプロトコルの 1 つである Solaris PPP の概要を示します。仕様、最も典型的な PPP 構成の紹介、PPP に関連した用語の定義について説明します。

Solaris PPP の概略

PPP を用いると、モデムと電話回線を使って、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 を使って、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ポイントツーポイントプロトコル (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 を用い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。

注 - x86 アーキテクチャのマシンには、一定の速度以上で実行される UART が必要です。

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、たとえばイーサネット、トークンリング、FDDI などにを使って 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、イーサネット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、7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 PPP 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 7–1 にこの概念を示します。

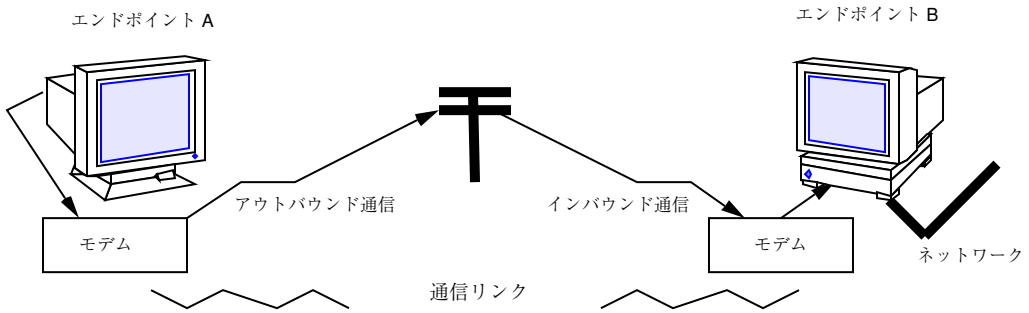


図 7–1　基本的なポイントツーポイントリンク

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動作と相手に渡す情報を、アウトバウンド通信と言います。

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言います。

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 7–1)
- 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 7–2)
- 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 7–3)
- 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 7–4)

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。以下の各節では、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 8 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 7–1 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われています。

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 7–2 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。

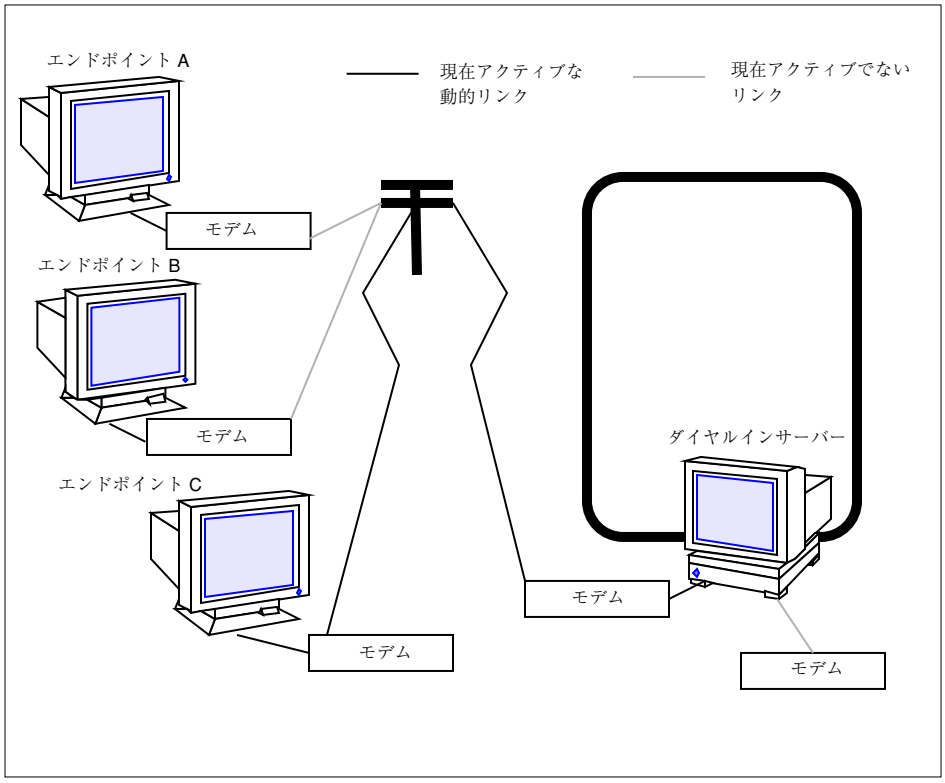


図 7–2　可搬コンピュータと動的リンクを持つダイヤルインサーバー

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 7–2 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、118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 7–2 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 7–1 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っ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 7–3 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。

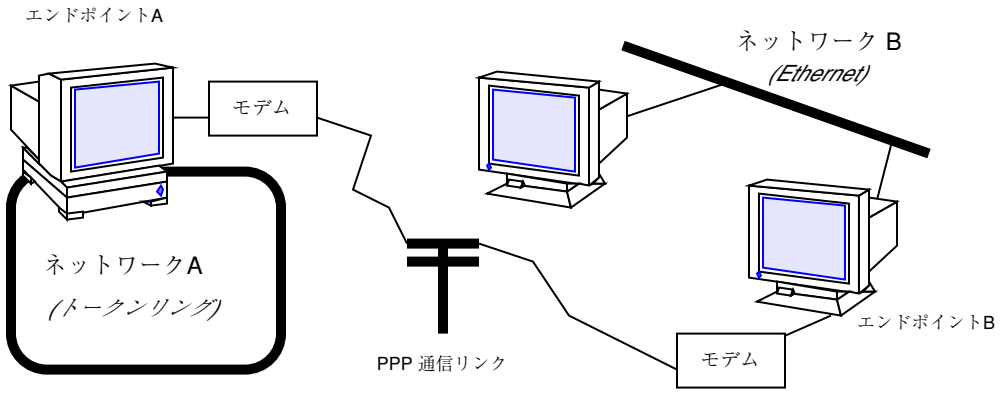


図 7–3　PPP リンクで接続された 2 つのネットワーク

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。

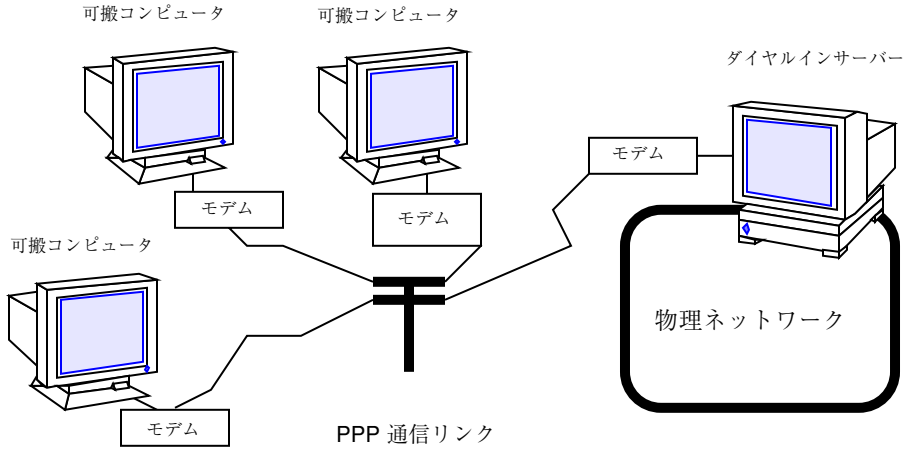


図 7–4　可搬コンピュータとマルチポイントダイヤルインサーバー

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。

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 7–4)
- 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 7–5)

以下の各節では、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 8 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 7–4 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 (117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 7–4 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。イーサネット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 7–5 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。

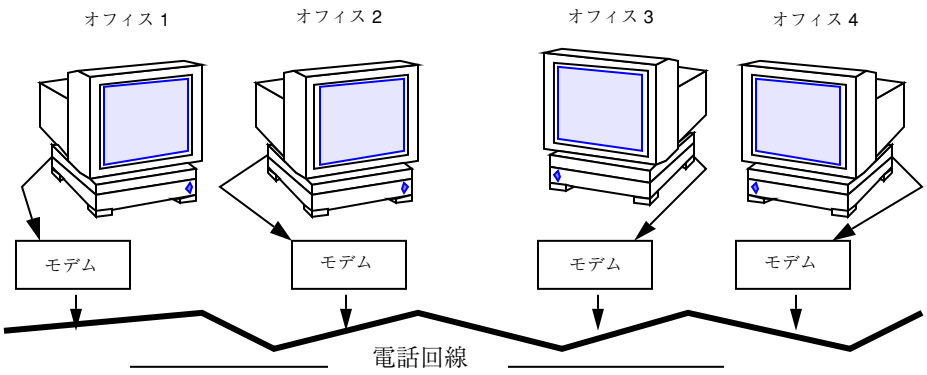


図 7–5　可搬コンピュータの仮想ネットワーク

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以下のものがあります。

- リンクマネージャ (`/usr/sbin/aspppd`)
- ログインサービス (`/usr/sbin/asppp1s`)
- 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 7–6 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。

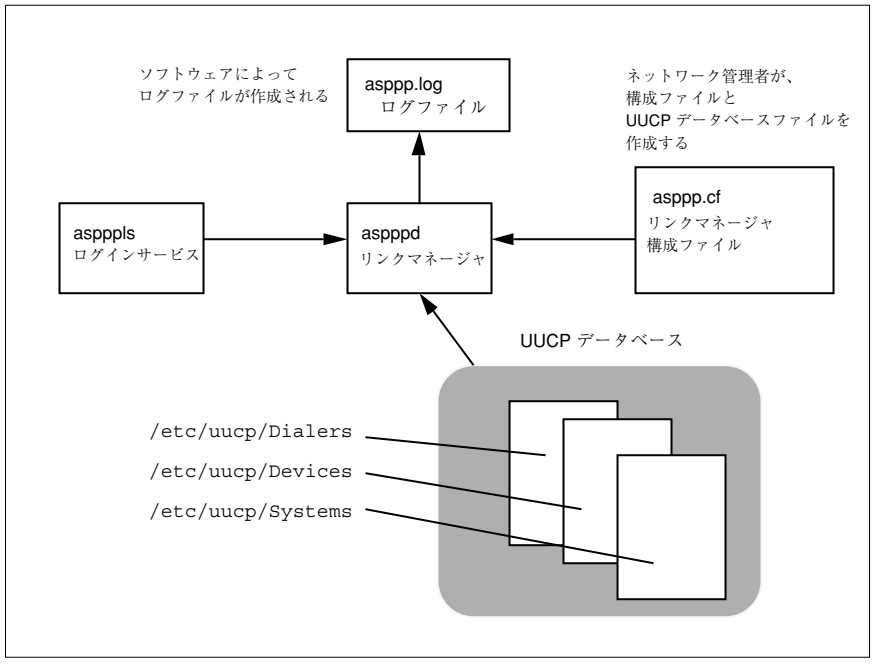


図 7–6　PPP のコンポーネントソフトウェア

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。

リンクマネージャについての詳細は、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、198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、150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。コード例 7–1 に、変更されていない asppp.cf ファイルを示します。

コード例 **7–1**　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、191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 asppp1s 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、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ほかに、/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、 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 12 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概要

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、以下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以下のことを行う。
   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。

その後、リンクマネージャは、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概要

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、以下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って、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。

その後、リンクマネージャは、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 をインストール後、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、185ページの「PAP/CHAP セキュリティのための asppp.cf の編集」で説明します。

第 8 章

PPP 構成の準備

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。

- ネットワークアドレス指定スキーマの決定
- ハードウェアが PPP の要件を満たしているかどうかの確認
- PPP の要件を満たすソフトウェアの準備

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。

構成に応じた要件の決定

Solaris PPP は、以下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、 第 7 章 の 114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うかどうか
- 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 PPP 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、134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、146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。パフォーマンス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、91ページの「ホストで静的ルーティングを選択するには」と、137ページの「RIP を不使用にする」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、134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` ファイルを使って、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。パフォーマンス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、91ページの「ホストで静的ルーティングを選択するには」と、137ページの「RIP を不使用にする」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、134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 ファイルを使って、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、134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。パフォーマンス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、91ページの「ホストで静的ルーティングを選択するには」と、137ページの「RIP を不使用にする」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。128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、134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、136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 ソフトウェアを使って、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、134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、135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、これはリンクのパフォーマンス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、第 9 章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、80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

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 1 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)
- イーサネット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、59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 3 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 8–1 は、PPP を用い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 8–1 PPP を用い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、リンクのパフォーマンス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、91ページの「ホストで静的ルーティングを選択するには」を参照してください。

RIP を不使用にする

/etc/gateways ファイルによって、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することができます。このファイルはオペレーティングシステムに付属しているものではないので、テキストエディタを使って作成する必要があります。

RIP を使用しないようにするには、/etc/gateways に次のエントリを入力する必要があります。

```
norip ipdptpn
```

ipdptp*n* は、使用するポイントツーポイント PPP インタフェースのデバイス名です。

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 2 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。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。

ファイルスペースの要件

以下のディレクトリ内に、PPP 用の十分なスペースを確保する必要があります。

- `/usr`
- `/usr/kernel/drv`
- `/usr/kernel/strmod`
- `/usr/sbin`

64 ビットの PPP 用には、以下のディレクトリ内に十分なスペースを確保する必要があります。

- `/usr/kernel/drv/sparcv9`
- `/usr/kernel/strmod/sparcv9`

PPP は、/usr に約 243 K バイト、/ (ルート) に約 4 K バイトを必要とします。

64 ビットの PPP は、32 ビットの PPP と同じサイズのディスク容量を必要とします。/usr/kernel/drv/sparcv9 に 64 ビットドライバ、/usr/kernel/strmod/sparcv9 に 64 ビットモジュールがあります。

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 8–1 PPP 構成前のチェックリスト

/usr に使用可能な空き領域が 300 K バイトありますか。	はい/いいえ
/ (ルート) に使用可能な空き領域が 4 K 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っ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* 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。	_____ _____ _____ _____
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 9 章

PPP の構成

この章では、PPP を構成するための手順と情報を記載しています。説明に使用する例は、リモートホストとそのマルチポイントダイヤルインサーバーの両方の種類の PPP リンクを持つ構成を想定しています。その他の種類の PPP 構成の設定方法については、第 11 章に記載されています。

構成プロセスの概要

第 8 章で述べたプリインストール作業が終われば、いよいよ 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。

インストールの確認

先へ進む前に、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。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。

```
SUNWpppk        # Contains kernel modules
SUNWapppu       # Contains the link manager and login service
SUNWappp        # Contains configuration files
```

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。

```
SUNWpppk        # Contains the 32-bit kernel modules
SUNWpppkx       # Contains the 64-bit kernel modules
SUNWapppu       # Contains the link manager and login service
SUNWappp        # Contains configuration files
```

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ってインストールしてください。

注 - pkgadd を使って PPP をインストールする場合は、上記のスクリーンボックスに並べた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 9–1 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 のほかに、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。

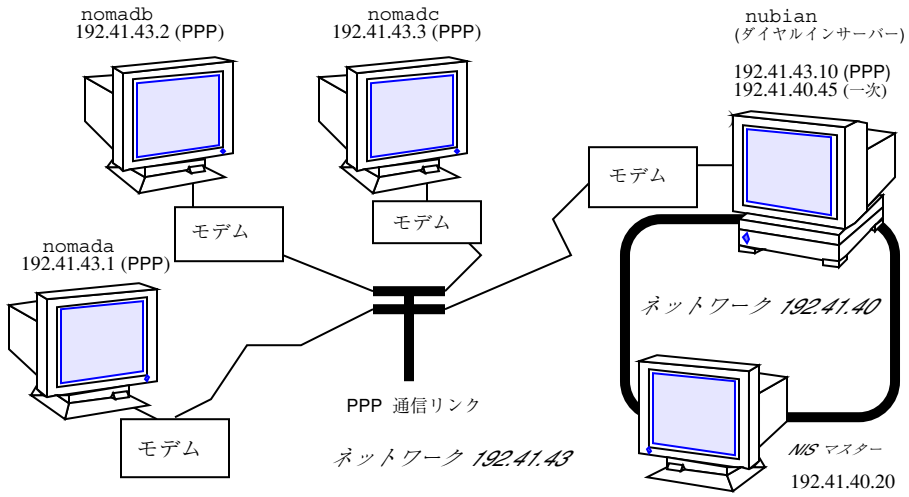


図 9–1　リモートホストとマルチポイントダイヤルインサーバーのネットワーク例

/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. スーパーユーザーとなり、/etc/inet/hosts ファイルを編集するための準備を整えます。

2. 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 9–1 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

入れられます。nomadb と nomadc の /etc/inet/hosts ファイルについても、同じことが行われます。

3. ダイヤルインサーバーの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
```

4. ネットワークで使用中のネームサーバーがある場合に、リモートホストのホスト名と **IP** アドレスによって、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. **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 9–1 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
```

2. 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。
3. 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。

136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 12 章で詳しく説明します。

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のであれば、/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、211ページの「/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
```

このエントリの最初のパラメータである usrv32bis は、/etc/uucp/Devices ファイルの最後のパラメータに対応しており、これによって両者が結合されます。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 12–6 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 *2* 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 9–1 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、203ページの「/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/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 9–1 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。

```
bin:x:2:2::/bin:
sys:x:3:3::/bin:
uucp:x:5:5::/usr/lib/uucp:
nuucp:x:9:9::/var/spool/uucppublic:/usr/lib/uucp/uucico
news:x:6:6::/var/spool/news:/bin/csh
sundiag:x:0:1:System Diagnostic:/usr/diag/sundiag:/usr/diag/sundiag/sundiag
lily:x:20:99:Dial-in Operator:/home/nubian/lily:/bin/csh
nomada:x:21:99:R. Burton:/:/usr/sbin/aspppls
nomadb:x:22:99:T. Sherpa:/:/usr/sbin/aspppls
nomadc:x:23:99: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って、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 (default セクションで使用するキーワードの説明については、第 11 章を参照してください)。

コード例 9–1 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。

コード例 9–1 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、第 10 章と、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)。コード例 9–1 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、152ページの「マルチポイントダイヤルインサーバーの構成ファイル」を参照してください。

コード例 9–1 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 ファイルを読んだ後で、/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。148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、第 11 章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。

コード例 9–2 は、図 9–1 に示したダイヤルインサーバー nubian の構成ファイルです。

コード例 **9–2**　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 ipd*n* 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。

コード例 9–2 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になります。

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、コード例 9–2 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、第 11 章に詳しい説明があります。

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

ファイル内のキーワード入力については、上記以外には形式上の必要条件はありません。

▼ asppp.cf 構成ファイルの編集方法

1. **1 つのエンドポイントマシンで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. 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。

3. 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。

4. 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。

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へ 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。185ページの「PAP/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。

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、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. ps コマンドを実行します。

```
# ps -e | grep asppp
```

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。

2. 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」のような出力が表示されるはずです。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。

3. 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、第 10 章 を参照してください。

▼ PPP の停止方法

1. ネットワーク上での **PPP** 操作を停止するには、次のように入力します。

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

第 10 章

PPP の障害追跡

この章では、ネットワーク上に PPP を構成した後に行う必要のある一連の検査事項を示します。その後、PPP リンクを介した通信に問題が生じたときには、PPP 診断機能を問題の解決に用いることができます。

要約すると、これらの検査は次の順序で行う必要があります。

1. ハードウェア
2. インタフェースの状態
3. 接続
4. ネットワークインタフェースの動作状態
5. ローカルルーティングテーブル
6. アクセス権
7. パケットフロー

PPP がすべてのテストに合格すれば、TCP と UDP のサービス、たとえば rlogin、telnet、ftp などを、リンクを介して使用できるようになっているはずです。それでもリンクに障害がある場合は、PPP 診断機能を利用して障害追跡を試みてください。

以下の各節では、上記の各検査について詳しく説明します。

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、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の検査

PPP を起動した後は、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。コード例 10–1 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。

コード例 **10–1**　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
        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、コード例 10-2 に示すような出力が得られます。

コード例 **10–2**　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、98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。

接続の検査

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。

```
# ping elvis
```

ここで、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、97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。

インタフェースの動作状況の検査

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。

```
# netstat -i
```

99ページの「netstat コマンド」 と netstat(1M) 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の検査

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。

```
# netstat -r
```

次に出力例を示します。

```
Routing tables
Destination   Gateway    Flags  Refcnt  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
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
```

ありうる宛先ネットワークのそれぞれについて、ルーティングテーブルエントリが 1 つずつあることを確認してください。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、適正な静的送信経路を追加します。in.routed を用いて動的ルーティングを使用しているときは、次のようにします。

1. 次のように入力して、in.routed が実行中であるかどうかを確認します。

   ```
   # 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。

2. 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
```

3. 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
```

4. 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 コマンドを使って、ネットワークからパケットを観察し、各パケットの内容を観察します。 コード例 10–3 に、snoop からの出力例を示します。

コード例 **10–3**　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` を使っ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、後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、イーサネット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 診断機能を用いた障害追跡

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用い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
```

aspppd.conf ファイルについての詳細は、149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、次のようにします。

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. 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.
3. 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。
4. 現在の aspppd デーモンを終了し、再起動します。

```
# kill PID
# aspppd
```

ここで、*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。

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 12 章を参照してください)。

下記の 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、203ページの「/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、211ページの「/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 ファイルの中から tb9600 というエントリを見つけ、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
```

その後の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)
```

下記のメッセージでは、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
```

下記のメッセージは、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。 コード例 10–4 は、サンプルの構成要求を示します。

コード例 **10–4**　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
```

以下に、コード例 10–4 に示した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

その後のいくつかの行は、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
```

下記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
```

下記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。その後のメッセージは、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
```

下記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
```

第 11 章

PPP リンクの調整

この章には、第 9 章で述べた基本リンクに比べて少々特殊な PPP リンクを構成するために必要な情報を記載しています。主に 2 つの種類の PPP リンクの構成方法について説明します。2 つの種類とは、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、仮想ネットワーク (これはマルチポイントリンクを使用します) です。章末には、asppp.cf 構成ファイルで使用できるすべてのキーワードのリストを記載しています。

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 7 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 11–1 に示す構成例に基づいて説明を進めます。

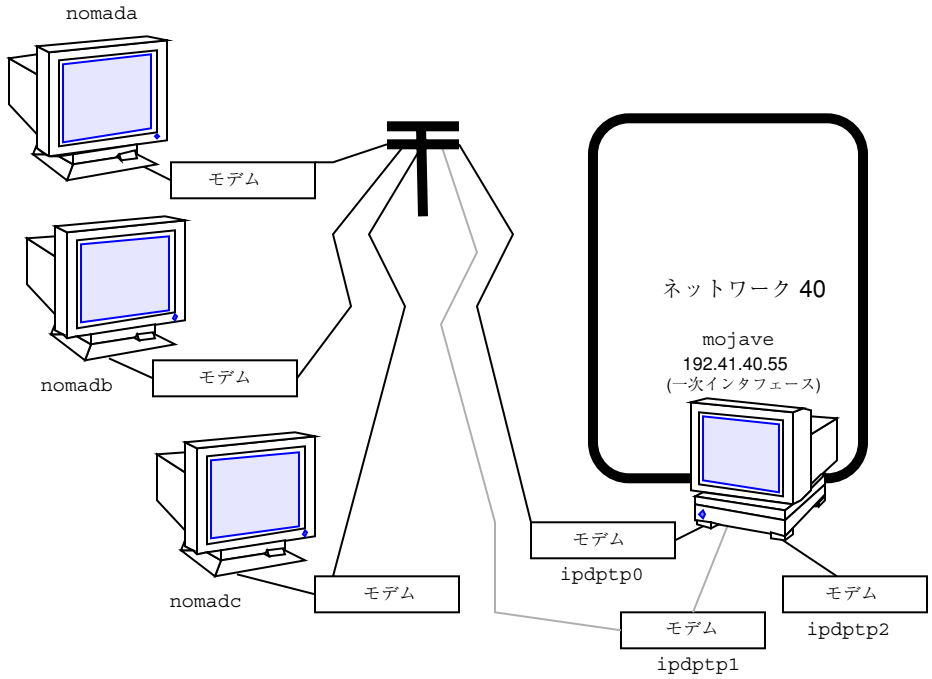


図 11–1　リモートホストと動的リンクダイヤルインサーバーのネットワーク

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ってダイヤルインサーバーと通信します。しかし、図 9–1 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、141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しかたも少々異なります。

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、明確にするには、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。

▼ 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. **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 11–1 では、nomada、nomadb、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。

2. 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
```

3. 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。

4. 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、次のようにします。

1. 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。

2. 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ついてのエントリを追加します。

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、148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っ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、150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どうかを示します。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。コード例 11–1 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、コード例 11–1 のようになります。

コード例 **11–1**　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

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

コード例 11–1 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
```

コード例 11–1 では、キーワード interface を使って 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、154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、上記のほかに、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、191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、119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。

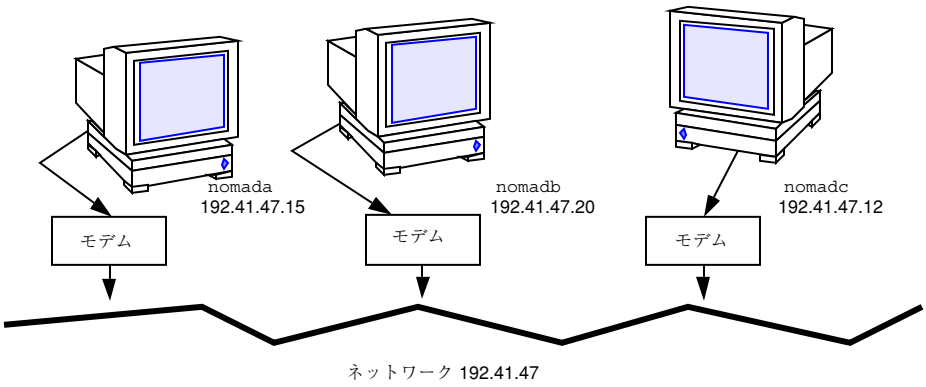


図 11–2　仮想ネットワーク例

図 11–2 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 (そしておそらくリモートロケーションの他のネットワーク管理者) 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、141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、136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 11–2 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 11–2 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、第 4 章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、146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、148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。

コード例 11–2 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。

コード例 11–2 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
```

コード例 11–3 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。

コード例 11–3 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/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、125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一般に認証を要求し、対等システムつまりクライアントは一般に認証を行います。

表 11–1 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 11–2 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	chap_secret
pap_password	chap_name

▼ PAP/CHAP のインストール方法

1. サーバーでスーパーユーザーになり、/etc/asppp.cf ファイルを編集する準備を整えます。

2. 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 11–3 を参照してください。また、コード例 11–4 は、/etc/asppp.cf ファイルの例を示します。

3. will_do_authentication キーワードを使って、リンク上で **PAP** 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。

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 11–3 を参照してください。また、コード例 11–4 は、/etc/asppp.cf ファイルの例を示します。

PAP/CHAP 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である。キーワードの構文は次のとおり

```
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、関連のパス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。

- 該当するパスについて、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。

表 11–3 PAP/CHAP のキーワードの定義

キーワード	値の定義
require_authentication *keywords*[1]	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*[2]	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[3] 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*[4]	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

表 11–3　PAP/CHAP のキーワードの定義　続く

キーワード	値の定義
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

1. キーワード として使用できるのは `off|pap[chap] | chap[pap]`

2. *peername* は、認証システムから見てポイントツーポイントリンクの反対側にあるシステムの名前です。これは、 4. に示す構文の文字列です。

3. オクテットはバイトの厳密な定義です。

4. *string* はホワイトスペースを含まない単一トークンです。特殊文字を含めるには、標準 ANSI の \ エスケープ文字を使用できます。空白文字を入れるには、\s を使用します。文字列の先頭にポンド記号がある場合は、コメントとして解釈されないようにするために、エスケープ (\#) する必要があります。NULL (\0) は文字列を切り捨てます。

PAP/CHAP の例

コード例 11–4 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。

コード例 11–4　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
```

コード例 11–5 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。

コード例 11–5 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
```

コード例 11–6 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。

コード例 11–6 リモートホスト nomadbe 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
```

一般に、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おうとするのが、理想的な形です。しかし、逆にリモートホストの方が認証を要求するように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。通常、CHAP シークレットは人間が直接先方に渡すという方法をとっています。

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、1334 を参照してください。

表 11–4 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 11–4 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、150ページの「asppp.cf ファイルの ifconfig セクション」、153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」、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 もしく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 11–4 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 11–5 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 11–5 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 層がダウン状態になったときは、この送信経路は削除される。
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。
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。

表 11–5 asppp.cf の省略可能キーワード 続く

キーワード	値の定義
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。本書では、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 。

パートIII　UUCP 通信の管理

UUCP ファイル転送システムを使用すると、UNIX ベースのシステム間でファイルと電子メールを送受信できます。パート III では、複雑な UUCP システムを管理する方法について説明します。

パート III の内容は、モデム管理と広域ネットワークに関する実践的な知識を持つ、熟練のネットワーク管理者を対象としています。また、ネットワーク管理者が、電話回線を介して UUCP を使用しようとしていること、コンピュータにハードウェアを追加する手順を熟知していること、すでにモデムをマシンに接続してあること、`tip` または `cu` を用いたダイヤルアウトができることを前提として説明を進めます。

第 12 章

UUCP のデータベースとプログラム

この章では、UUCP のプログラムとデーモンを紹介します。その後で、UUCP 構成の一環として、UUCP データベースファイルを構成する方法について詳しく説明します。データベースの作成後、UUCP をどのように構成するかについては、第 13 章で説明します。

UNIX-to-UNIX Copy Program (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 を使って 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ます。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 15 m 以内の場合は、直接リンクを使用すると便利です。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ます。
- 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ます。モデムは、UUCP が要求する電話番号をダイヤルします。受信側のモデムは、着信に応答できなければなりません。
- 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ます。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ます。

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 *2* 巻*)*』と、モデムに付属している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` を使って 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。

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します。uucico は、UUCP ログインアカウント用の「ログインシェル」として働きます。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します。

  uucp、uuto、uux の各プログラムは、必要なファイルをすべて作成してから、uucico デーモンを実行して、リモートコンピュータに接続します。uusched と Uutry は、どちらも uucico を実行します (詳細は uucico(1M) のマニュアルページを参照してください)。

- uuxqt – リモート実行要求を処理します。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*) を見つけます。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します。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します。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します。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ます (詳細は uuxqt(1M) のマニュアルページを参照してください)。

- 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します。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ます (詳細は uusched(1M) のマニュアルページを参照してください)。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します。

- in.uucpd – ネットワークを介した UUCP 接続をサポートします。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 を呼び出します。次に、uucpd がログイン名を要求します。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りません。次に in.uucpd は、不要な場合を除いてパスワードを要求します (詳細は in.uucpd(1M) のマニュアルページを参照してください)。

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 を用い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します。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ます。ログファイルには、uucp、uuto、uux の使用が記録されます (詳細は uucp(1C) のマニュアルページを参照してください)。
- uucleanup – スプールディレクトリをクリーンアップします。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ます (詳細は uucleanup(1M) のマニュアルページを参照してください)。
- 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ます。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します (詳細は Uutry(1M) のマニュアルページを参照してください)。
- uucheck – UUCP のディレクトリ、プログラム、サポートファイルの有無を検査します。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します (詳細は uucheck(1M) のマニュアルページを参照してください)。

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します。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ます (詳細は cu(1C) のマニュアルページを参照してください)。
- uucp – あるマシンから別のマシンへファイルをコピーします。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 デーモンを呼び出します。このデーモンは、リモートコンピュータへの接続を試みます (詳細は uucp(1C) のマニュアルページを参照してください)。
- 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します。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 を使ってそのファイルを取り出すよう指示します (詳細は uuto(1C) のマニュアルページを参照してください)。

- uupick – uuto を用い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します (詳細は uuto(1C) のマニュアルページを参照してください)。
- 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、実行ファイルを作成します (詳細は uux(1C) のマニュアルページを参照してください)。
- uustat – 要求された転送 (uucp、uuto、uux) の状態を表示します。また、キューに入っている転送を制御する手段も提供します (詳細は uustat(1C) のマニュアルページを参照してください)。

UUCP データベースファイルの紹介

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以下のものがあります。

- Config – 変数パラメータのリストが入っています。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ます。
- Devconfig – ネットワーク通信を構成するために使用されます。
- Devices – 自動呼び出し装置 (モデム)、直接リンク、ネットワークデバイスの位置と回線速度に関する情報が入っています。これは、UUCP のほかに PPP でも使用されます。
- 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ます。これは、UUCP のほかに PPP でも使用されます。
- 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ます。これは必須ではありませんが、UUCP のほかに PPP でも使用できます。
- 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します。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ます。

- Limits – このマシンで同時に実行できる uucico、uuxqt、uusched の最大数を定義します。
- 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します。
- 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します。
- Sysfiles – uucico と cu が、Systems、Devices、Dialers ファイルとして、別のファイルや複数のファイルを使う時に、その割り当てを行います。
- Sysname – TCP/IP ホスト名のほかに、各マシンに固有の UUCP 名を定義できます。
- Systems – uucico デーモン、cu、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ます。この情報には、リモートホストの名前、リモートホストに対応する接続デバイスの名前、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ます。

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ほか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 ファイルの構成設定

UUCP データベースは、201ページの「UUCP データベースファイルの紹介」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。一般に、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。201ページの「UUCP データベースファイルの紹介」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。

この章の以降の部分では、UUCP データベースについて詳しく説明します。

/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、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 12–1 に、Systems ファイルのフィールドの例を示します。

例 **12–1**　/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
```

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。226ページの「/etc/uucp/Sysname ファイル」を参照してください。例 12–1 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。

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、以下のエントリのいくつかを含むリストを指定できます。

表 **12–1**　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 12–1 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。

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。

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、例 12–2 に示すように、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 (表の見出しに示されているフィールドは Systems ファイル用のものであり、Devices ファイルには適用されないので、注意してください。Devices ファイルのフィールドの対応関係については、例 12–6 を参照してください)。

例 **12–2** 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
```

(続く)

```
Device    ACUEC        cua/a -        38400  usrv32bis-ec
```

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ています (プロトコルの詳細は、218ページの「Devices ファイル内のプロトコル定義」を参照してください)。

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 214ページの「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、例 12–3 に示したように、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 **12–3** 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
Device    ACU          tty11 --       D1200  penril
```

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。

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、例 12–4 に示すように Dialcodes ファイル内に列挙されているもののひとつでなければなりません。

例 **12–4** 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

Dialcodes NY 1-1212
```

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 ファイルを使って解釈されないようにしなければなりません。

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。

```
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 12–2 に、便利なエスケープ文字をいくつか紹介します。

表 12–2 `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

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

表 12–2　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

文字	説明
\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
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用いたコールバックの有効化

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。

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(1V) 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。

```
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。

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。

```
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。

/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。

Devices ファイルのエントリの形式は次のとおりです。

Type	*Line*	*Line2*	*Class*	*Dialer-Token-Pairs*

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。

```
ACUEC cua/a - 38400 usrv32bis-ec
```

以下各フィールドについて説明します。

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 219ページの「/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 12–5 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。

例 12–5 で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 203ページの「/etc/uucp/Systems ファイル」を参照してください)。

例 **12–5** Type フィールドと /etc/uucp/Systems の対応関係

```
File Name  Type   Line  Line2  Class  Dialer-Token-Pairs

Devices    ACUEC  cua/a -      38400  usrv32bis-ec

System     nubian Any   ACUEC  38400  9998888 '''' \d\d\r\n\c-ogin-\r\n\c-ogin.......
```

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
```

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。

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。

例 12–6 に示すように、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。各列の見出しは、Devices ファイルのフィールドのものであることに注意してください。

例 **12–6** Class フィールドと /etc/uucp/Systems の対応関係

```
File Name  Type   Line  Line2  Class  Dialer-Token-Pairs
Devices    ACU    cua/a -      D2400  hayes
System     gobi   Any   ACUEC  D2400  3251 ogin: nuucp ssword: taheya
```

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 2400 bps となります。

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 12–3 に、特殊ダイヤラタイプを示します。

表 12–3 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 を使わないもの)
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 を使うもの)

詳細は、218ページの「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、例 12–7 に示すように、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 (各列の見出しは、Devices ファイルのフィールドの見出しです)。

例 **12–7**　Dialers フィールドと /etc/uucp/Dialers の対応関係

```
File Name Type  Line  Line2 Class Dialer-Token-Pairs

Devices   ACU   cua/b -     2400  hayes

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、ダイヤラに渡すトークン (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 12–9 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 (212 ページの「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。例 12–8 に示すように、*dialer* 部が Dialers ファイルのエントリと突き合わされます (各列の見出しは、Devices ファイルのフィールドの見出しです)。

例 **12–8**　Dialers フィールドと /etc/uucp/Dialers の対応関係

```
File Name  Type     Line  Line2  Class  Dialer-Token-Pairs

Devices    develcon cua/a -      1200   develcon

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 (203ページの「/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。例 12–7 に示すように、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 (各列の見出しは、Devices ファイルのフィールドの見出しです)。

例 **12–9** Dialers フィールドと /etc/uucp/Dialers の対応関係

File Name	Type	Line	Line2	Class	Dialer-Token-Pairs		
Devices	ACU	cua/b	-	1200	develcon	vent	ventel
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 (トークン) フィールドを、/etc/uucp/Dialcodes ファイルを使っ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 (トークン) フィールドを、/etc/uucp/Dialcodes ファイルを使っ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。

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 (203ページの「/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 12–4 に、Devices ファイルで使用できるプロトコルを示します。

表 12–4 /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、最もよく使われる。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
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。

/etc/uucp/Dialers ファイル

/etc/uucp/Dialers ファイルには、よく使わ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、一般に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われます。

211ページの「/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 12–10 は、US Robotics V.32bis モデム用のエントリの例を示しています。

例 12–10 /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 12–11 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 12–11 /etc/uucp/Dialers ファイルの抜粋

```
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

#   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 12–5 に、Dialers ファイルの send 文字列でよく使われるエスケープ文字を示します。

表 12–5 /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
\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、203ページの「/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

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(1V) 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。

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、 Dialers ファイルに加えて使用できるファイルです。

/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 12–6 に、この 2 つのファイル間の対応関係を示します。

表 12–6 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 12–7 に示すのは、Dialcodes ファイルのエントリの例です。

表 12–7 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。

/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 コマンドを使って表示でき、cu リストは uuname –C コマンドを使って表

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
```

/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。

/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、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わない限り、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。

考慮事項

Permissions ファイルを使って、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、そのサイトには以下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)。

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。

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 つを使っ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 203ページの「/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。

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、233ページの「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。

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ほかに /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。

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。

CALLBACK オプション

LOGNAME エントリの中で CALLBACK オプションを使うと、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。

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、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。

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、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われる 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って、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、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。

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。

MACHINE と LOGNAME 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、WRITE オプションを共有しています。この 2 つを組み合わせ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。

転送

一連のマシンを介してファイルを送信するときは、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 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、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。

/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。

/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 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
```

/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って、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。

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。

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。Restrictions 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。

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 12–8 に示すオプションを使用できます。

表 12–8 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	キロバイト数を表す 10 進数 (K はキロバイトの略号)
*n*M	メガバイト数を表す 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

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 12–9 に、キーワードとそれぞれの意味を示します。

表 12–9 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のグループ名

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のキューイングが許可、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のキューイング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。

/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われるものです。

```
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。

/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、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、小文字を使っ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われる内容です。

```
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。

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。

管理ファイル

以下、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の名前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.*sysnxxxx* 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 12–10 に、UUCP ロックファイルの種類を示します。

表 12–10　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の名前を表す
LCK.*dev*	*dev* はファイルを使用しているデバイスの名前を表す
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。作業ファイルの名前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の名前、*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。データファイルの名前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の名前の最初の 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。実行ファイルの名前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の名前で、*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の名前
  - 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

第 13 章

UUCP の構成と保守

この章では、マシンに関連したデータベースファイルを変更した後で、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 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 接続を介してローカルシステムにアクセスすることを許可されているリモートマシンについては、次の例に示すようなエントリを /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。

上記のエントリは、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。

同様に、ローカルマシンの管理者も、ローカルマシンの名前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、ファイル uudemon.crontab の適切な行のコメントを解除してください。

uudemon.crontab ファイルを有効にするには、スーパーユーザーになって次のように入力します。

```
# su uucp
# 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。

1. 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
2. 結果の状態情報を、メールにより 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。

1. /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
2. 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
3. 配達できなかったメールを送信元に戻す
4. 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

/etc/inetd.conf 中で UUCP を有効にする

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。

/etc/inetd.conf の中で、次のエントリの前にコメントマーク (#) が付いていないことを確認します。

```
uucp stream tcp nowait root /usr/sbin/in.uucpd in.uucpd
```

TCP/IP 用に Systems ファイルエントリを修正する

/etc/uucp/Systems ファイルのエントリには、次のフィールドがあります。

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。

UUCP のための /etc/inet/services の検査

次に示す /etc/inet/services のエントリは、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。しかし、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。

セキュリティ、保守、障害追跡

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、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。

定義する各マシンについて、以下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。247ページの「uudemon.poll シェルスクリプト」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/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後で、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。

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。

障害のあるモデムや ACU の検査

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。

- `uustat –q` を実行する。接続障害のカウントと理由が示される
- `cu –d –l`*line* を実行する。*line* は `/dev/cua/a` である。これによって、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できる。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らない (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必要がある。あるいは、デバイスを `direct` として設定する必要がある)。

/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かどうかを確認してください。次のようなマシンに関する情報が、最新でなくなっている可能性があります。

- 電話番号
- ログイン
- パスワード

伝送のデバッグ

特定のマシンに接続できない場合は、`Uutry` と `uucp` を使って、そのマシンに対する通信を検査できます。

1. 接続を調べるために、**`/usr/lib/uucp/Uutry -r`** *machine* を入力し、**Return** を押します。

   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ます。

   b. デバッグ出力を `/tmp/`***machine*** に送る。

   c. 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 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 を使っても問題の原因が分からない場合は、**uucp -r** ***file machine*\!*/dir/file*** と入力し **Return** キーを押すことによって、ジョブをキューに入れてみます。

   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. ここで、再度 Uutry を使用します。

   それでも問題が解決できないときは、ご購入先への連絡が必要になります。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。

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。

プロセスがアボートされた場合は、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、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、一般に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

以下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 – このコマンドは、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 – このコマンドは、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 – このコマンドは、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。

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 13–1 に ASSERT エラーメッセージをリストします。

表 13–1 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 。
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 。

表 13–1 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 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。これが起こるのは、シーケンスファイルのアクセスに問題がある場合である。これは一般にソフトウェアエラーを示す。
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 。
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 。

表 13–1 ASSERT エラーメッセージ 続く

エラーメッセージ	説明と処置
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	おそらくリモートマシンがハングアップした。処置は不要。
PKXSTART	リモートマシンが回復不可能な状態でアボートした。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れかに問題がある。この処理の試行の前に宛先のモードが検査されていると考えられるので、スプールディレクトリに問題がある可能性が高い。
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 13–2 に一般的な STATUS エラーメッセージを示します。

表 13–2 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。

表 13–2 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 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、きわめて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、プログラムがアボートしたか、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ことが考えられる 。
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って、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、254ページの「UUCP のエラーメッセージ」を参照してください。
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してください。
DEVICE FAILED	デバイスがオープンできない
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を示している。
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。

表 13–2 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
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 を使用していないことが考えられる。
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 を使って検査してください)。

UUCP の数値エラーメッセージ

表 13–3 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 13–3 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。

表 13–3 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
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 (この種のエラーが捕捉できるようになっている場合)。
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 `/etc/utmp` などのシステムファイルのどれかが、存在しないか、オープンできない。あるいは、構文エラーなどのようなエラーがある。
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。
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。

表 13–3 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
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	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。

パートIV　動的なホスト構成プロトコル

動的なホスト構成プロトコル (DHCP) を使用すると、ホストはインターネットプロトコル (IP) アドレスおよび他のインターネット構成パラメータを取得することができ、ユーザーによる事前定義の必要がなくなります。この新しいプロトコルは、システム管理者が各 IP アドレスを個別に割り当ておよび変更を行う必要があった、従来のインターネットアーキテクチャを改善します。

DHCP を使用すると、管理者が何度も IP アドレスを割り当てたり、変更したりする必要がなくなるため、ネットワークの管理費用が削減されます。

- 第 17 章「DHCP の障害追跡」
- 306ページの「方法および注意事項」
- 313ページの「DHCP サーバーの障害追跡」
- 322ページの「DHCP クライアントの障害追跡」

第 14 章

DHCP の概要

この章では、動的なホスト構成プロトコル (DHCP) を紹介し、クライアント側とサーバー側の両方のプロトコルについて説明します。また、DHCP の動作内で使用できるブートストラッププロトコル (BOOTP) の中継エージェントについても説明します。

DHCP とは何か

DHCP は、インターネットプロトコル (IP) アドレスやその他のインターネット構成パラメータをホストに提供します。DHCP を使用すると、ユーザーが事前に構成を行う必要がなくなります。この新しいプロトコルにより、システム管理者が個々の IP アドレスを個別に割り当てたり変更したりする必要があった、従来のインターネットアーキテクチャが改善されます。手作業による処理には、費用がかかる、難しい、エラーが発生しやすい、時間がかかる、という短所があります。

DHCP は、管理者が何度も IP アドレスを割り当てたり変更したりする必要性をなくすことにより、ネットワークの管理費用を削減します。使用されていない IP アドレスのプールから動的 IP アドレスを選択して、一時使用または永久使用のために自

動的にホストに割り当てます。さらに、割り当てた IP アドレスが不要になるまたは使用期限が切れた場合には、DHCP はその他のクライアントに使用させるために当該 IP アドレスを回収します。

DHCP と BOOTP のパケットフォーマットは同じですが、BOOTP パケットは固定長、DHCP パケットは可変長です。DHCP パケットの長さについては、クライアントとサーバーとの間でネゴシエーションが行われます。

インターネット技術タスクフォース (IETF) の動的なホスト構成のための作業グループは、約 5 年間にわたって、現在の IP アドレス割り当てのアーキテクチャが持っている問題について作業を行なってきました。IETF は、DHCP の標準化を行なっている途上です。現状は、複数のコメント要求 (RFC)、すなわち RFC 1542、2131、2132 が出されている状態です。

インターネットが急速に成長しているため、ネットワークアドレスが不足しています。この問題に対処するために、クラスレスドメイン間ルーティング (CIDR) が開発されました。IP アドレスは、大規模、中規模、小規模のネットワークに対応するクラス A、B、C に分けられていました。クラス B の IP アドレスが不足したため、CIDR の設計が使用されることになりました。CIDR は、65,536 個のアドレスから構成されるクラス B ネットワークを 1 つの組織に 1 つ割り当てるのではなく、必要な数だけのクラス C アドレスを割り当てるという考えをベースにしていました。

CIDR 戦略に従って割り当てられたクラス C ネットワーク番号は、ランダムではありません。番号は連続しており、同じ接頭辞を共有しています。このことが、非常に大きいルーティングテーブルを操作することによって引き起こされる問題の大部分を軽減するのに役立ちます。

以前は、IP アドレスのブロックが個別の要求者や企業に割り当てられていましたが、CIDR 戦略の場合は、個別の ISP に割り当てられます。したがって、ISP を簡単に変更するためには、簡単に番号の付け直しができることが重要になります。DHCP では、ネットワークの番号を簡単に付け直すことができ、したがって、ISP を簡単に変更することができます。

DHCP の技術では、以下の便利な機能を使用することができます。

- 自動的なネットワーク構成。これにより、DHCP をサポートするネットワーク上でクライアントマシンに電源を入れると、クライアントマシンは必要な情報を獲得することができます。たとえば、IP アドレス、デフォルトのルーター、サブネットマスク、DNS ドメイン、DNS サーバー、NIS+ ドメイン、NIS+ サーバー、タイムゾーン、時間サーバー、および動作に必要なその他のすべての情報を獲得しますが、管理者が個別に構成を行う必要はありません。

- 自動的な IP アドレス管理、すなわち、IP アドレスの割り当て、回収、再構成、再番号付け。TCP/IP の全パラメータは集中管理され、かつ DHCP サービスが IP アドレスの割り当てと再使用を自動的に管理するため、管理者の関与の必要性は最小限です。
- ネットワーク構成情報の位置が集中化されたことによって非常に簡素化されたシステム管理。TCP/IP 構成情報は、ネットワーク内のクライアントのそれぞれに分散するのではなく、中央の 1 か所に格納することができます。1 つのネットワークの番号の付け直しは、相対的に短期の IP アドレスのリースを使用して、簡単かつ迅速 (通常は数日以内) に行うことができます。

さらに、DHCP は BOOTP をベースとしているため、既存の BOOTP 中継機能を活用することができます。これにより、ネットワーク管理者は、ルーターを設定して BOOTP/DHCP トラフィックをリモート BOOTP/DHCP サーバーへ転送することができます。その結果として、`in.rarpd` と `bootparams` による構成サービスの場合には必要であった、ネットワークセグメントごとのネットワークパラメータサーバーは必要なくなりました。

ネットワーク管理者は、このコマンドを使用して、Solaris サーバー上に DHCP サービスと BOOTP サービスを迅速かつ容易に構成することができます。このコマンドは、DHCP サービスを立ち上げて、ローカルネットワークとリモートネットワークを構成します。ローカルネットワークは、サーバーが直接接続されているネットワークです。リモートネットワークは、サーバーが直接には接続されていないネットワークであり、BOOTP 中継エージェントを介してアクセスします。

リモートネットワーク上のクライアントにサービスを提供するには、クライアントのネットワーク上で BOOTP 中継エージェントを設定する必要があります。BOOTP 中継エージェントの機能は、普及している数多くのルーターや交換機内に存在します。ルーターがこの機能をサポートしていない場合は、クライアントのネットワーク上の任意の Solaris マシン上で、`in.dhcpd` デーモン (`in.dhcpd(1M)` マニュアルページを参照) を中継エージェントモードで動作させることができます。これは、Solaris 2.6 から利用できるようになった方法です。図 14–1 に、DHCP と BOOTP の概要を示します。

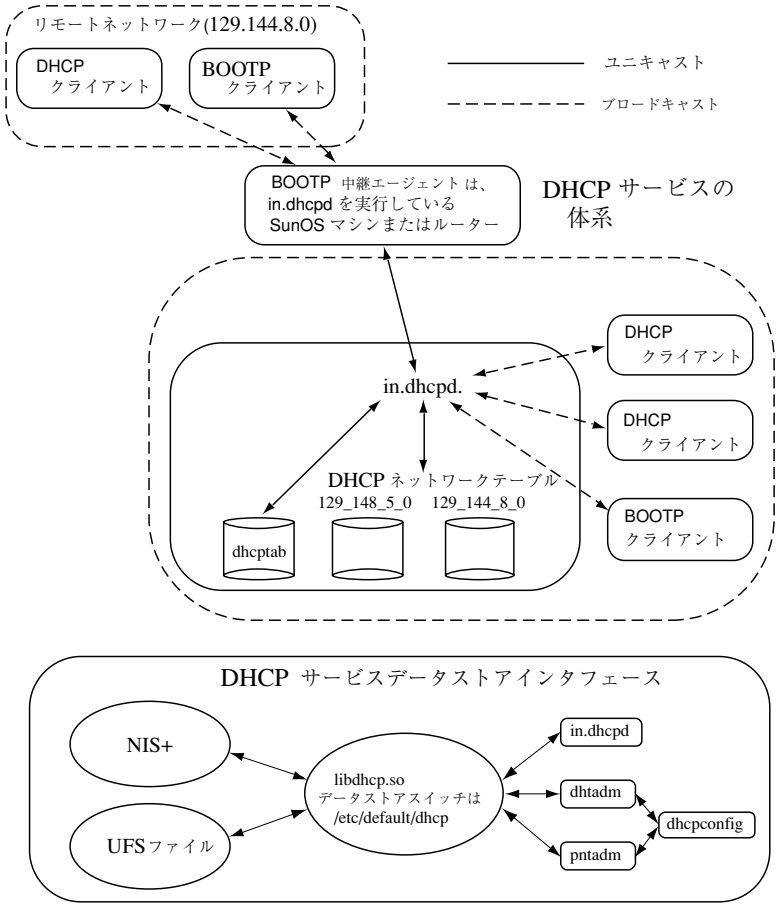


図 14–1 DHCP アーキテクチャ

DHCP クライアント

DHCP プロトコルには、クライアントに関連する 2 つの機能があります。1 つは、クライアントがネットワーク通信の終端を確立することができるように、クライアントに対して十分な情報を配信することです。もう 1 つの機能は、システムレベルとアプリケーションレベルのソフトウェアに必要な、その他のパラメータを提供することです。

クライアント情報の配信

最初の機能を実行するにあたり、DHCP プロトコルはクライアントのハードウェアインタフェースに接続されているネットワークに対して有効な IP アドレスを提供します。IP アドレスを使用する権利は設定された期間だけ与えられ、これをリースと呼びます。この点が、従来の静的構成と異なります。クライアントが、元のリースよりも長い期間に渡ってこの IP アドレスを使用したいと希望する場合は、サーバーとの間で DHCP を介してリースの延長についてネゴシエーションを定期的に行う必要があります。クライアントに IP アドレスが不要になった場合、マシンのユーザーはリースを放棄して、使用可能な IP アドレスのプールへ IP アドレスを戻すことができます。ユーザーが戻さない場合、その IP アドレスはリースの期限が切れると自動的に回収されます。

クライアント側の DHCP プロトコルを Solaris 上に実装する場合には、複数の条件を満たす必要があります。サンのワークステーションを起動することは、構成や起動が必要なサービスが多様でかつその数が多いため、複雑なプロセスです。DHCP による解決策はすべて、すでに使用されているその他の方法 (特に、逆アドレス解決プロトコル (RARP) および静的な構成) と併存する必要があります。ワークステーションの起動後、スーパーユーザーがネットワークのアドレスを変更することができることを認識できる必要があります。複数のインタフェースが構成できる必要があります。人間による制御に応答する必要があり、さらにプロトコルの状態についての報告と統計を提供できる必要があります。

Solaris DHCP クライアントは、複数の機能を実装することによってこれらの条件を満たします。最初の起動から数日または数週間後にはリースを更新する必要があるため、DHCP を受け持つエージェントを、クライアント上でデーモンとして動作させる必要があります。このデーモン、すなわち DHCP エージェントあるいは `dhcpagent(1M)` (`dhcpagent(1M)` のマニュアルページを参照) は、プロトコルの対話をすべて受け持ちます。デーモンは、サーバーへの接続の際には、DHCP プロトコルの全パケットの送受信を行います。このデーモンは、以下のことを実行します。

- パケットを構築して送信します
- サーバーからの応答を待機します
- 受け取った構成情報をキャッシュに書き込みます
- リースを解放または更新します
- インタフェースを介したネットワークとの通信を可能にするのに十分な情報を用いて、インタフェースを構成します。

エージェントの役割はこれですべてです。クライアントが動作している可能性がある、より上位のレベルについては、このデーモンはまったく関知しません。

開始時点では、エージェントは DHCP がどのインタフェースを構成するかについては何も想定しないで他のエンティティからの命令を待機します。これらの命令は制御プロトコルによってエージェントへ引き渡されますが、さらに、この制御プロトコルによって状態とその他の情報がエージェントからコントローラへ戻されます。ユーザーはこのコントローラによってエージェントの動作を制御することができ、エージェントの動作の制御は、このコントローラによって `ifconfig(1M)` コマンドの新規機能を介して実現されます。

`ifconfig` コマンドには、インタフェース上での DHCP の開始や終了を行う、DHCP 専用の新しいコマンド行オプションがあります。DHCP が開始されると、エージェントはプロトコルの命令に応じてサーバーとの間でパケットの送受信を行います。

最も簡単な例として、DHCP によってインタフェースが正常に構成された場合を考えます。エージェントは、リースの存続期間を記録し、インタフェースが構成済みである旨を `ifconfig` に通知し、受け取った構成をディスクに書き込んで休眠状態になります。

事前に設定された将来のある時点 (通常はリース存続期間を 50% 経過した時点) になると、エージェントは再び休眠状態から覚めて、リースの延長についてネゴシエーションを行います。このネゴシエーションは、ワークステーションが動作している間は何回でも無制限に行うことができます。最後には、システムを停止する時が来ます。その場合は、`ifconfig` によってエージェントに通知して、リースを放棄することができます。リースを放棄すると、ディスク上に格納されている構成情報はもはや有効でなくなるため削除されます。

エージェントは、数多くの別々のインタフェースを同時に追跡することができますが、それらのインタフェースの更新を同時に行う必要はありません。

上記の例は最も単純な場合ですが、状況がより複雑な場合も考えられます。エージェントが、自己のプロトコルメッセージに対する応答をまったく受け取らない場合もあります。その場合、エージェントはディスクに格納されていた構成を使用することができます。ただし、それが可能なのは、その構成に関連付けられたリースの期限が切れていない場合に限られます。有効な構成が見つからない場合、エージェントは定義済みの再伝送スケジュールを使用して DHCP を再試行を継続することもできますが、インタフェースの構成に失敗することもあります。どちらになるかは、インタフェースが主インタフェースとして指定されているかどうかに依存します。インタフェースが主インタフェースとして指定されている場合、エージェントは DHCP の再試行を継続します。インタフェースが主インタフェースとして指定されていない場合、エージェントは `ifconfig` コマンドに失敗します。

さらにエージェントは、人間の介入が発生した可能性も考慮する必要があります。エージェントが休眠状態から覚めた際に、IP アドレスとインタフェースの状態が受け取った構成と一致していないことを発見した場合、エージェントはそのインタフェースをアクティブリストからはずします。リースの取得がエージェントに対して再度要求されるまでは、インタフェースに対しては DHCP 操作は何も発生しません。

追加情報の提供

2 番目の機能、すなわちアプリケーションレベルとシステムレベルの情報の配信を実行するにあたり、Solaris DHCP クライアントは別のプログラムの `dhcpinfo(1)` を使用します。エージェントはこれらのサービスについては何も知らないため、`dhcpinfo` が取り出すのを待機し、DHCP プロトコルを介して受け取った構成情報をすべて格納します。

`dhcpinfo` コマンドは、指定されたパラメータを用いてコマンド行の引数を解釈し、当該パラメータの値についてエージェントに問い合わせ、その結果を (人間が読める) テキストの文字列として標準出力に表示します。ただし、`dhcpinfo` の応答を主に使用するのはユーザーではなく Solaris の起動スクリプトです。シェルコマンドの置換や出力の切り替えの際にこの出力を容易に使用することができるからです。

DHCP が提供するデータは、ホスト全体についてのデータもあれば、インタフェース専用のデータもあります。DHCP が構成可能なインタフェースを 1 つだけ持つクライアントではこの違いは意味がありませんが、数多くのインタフェースをもつホストでは `dhcpinfo` パラメータに関する解釈の疑問が生じます。たとえば、エージェントが 2 つのインタフェースを構成可能であり、かつこの 2 つのインタフェースに対して戻された NIS+ ドメイン名が異なる場合があります。この状況は、インタフェースを 2 つのカテゴリ、すなわち主インタフェースと二次インタフェースに分けることによって解決することができます。

主インタフェースは、ホスト全体の構成の場合に優先されるインタフェースです。`dhcpinfo` は、値を尋ねられると主インタフェースに問い合わせます。インタフェース固有のデータの場合も同じように行われます。戻される値は、主インタフェースついて受け取ったデータになります。たとえば `dhcpinfo` は、IP アドレスを尋ねられると主インタフェースの IP アドレスを標準出力に表示します。`ifconfig` に対するコマンド行の引数によって、インタフェースを主インタフェースとして指定します。

`dhcpinfo` コマンドでは、その他のコマンド行オプションによってデフォルトの動作を無効にすることができます。それらのオプションのうちの 1 つを使用する

と、インタフェース名を明示的に指定することができます。その場合、戻される値は当該インタフェースに対して DHCP が配信した値になります。

ホスト全体についてのデータの大部分は、Solaris クライアントを正常にブートするために非常に重要なので、1 つのインタフェースを主インタフェースとして指定すると、エージェントがその主インタフェースを構成できない限りシステムがブートできません。dhcpagent が主インタフェースの構成を完了するまで無期限に待機するように、コマンド行の引数によって ifconfig コマンドに命令します。

DHCP サーバー

DHCP サーバーは、そのサーバーに直接接続されているネットワークの IP アドレス空間を管理します。この環境をその他のネットワークに拡張する場合は、DHCP サーバーまたは BOOTP 中継エージェントをそれらのネットワーク上で構成する必要があります。

DHCP サーバーは、主サーバーとしても二次サーバーとしても動作することができます。主サーバーであるためには、一定の範囲の IP アドレスを受け持つ必要があります。

注 - 主という用語は、クライアントとサーバーでは異なる意味で使用されます。

すでに主 DHCP サーバーが存在するネットワークに DHCP サーバーを追加する場合は、新規サーバーを、主サービスと二次サービスを提供するように構成することも、また二次サービスだけを提供するように構成することもできます。新規サーバーを両方のサービス用に構成した場合は、それぞれが異なる IP の範囲を受け持っている限り、両方のサーバーが主サーバーの役割を実行することができます (IP アドレスを配付することができます)。確認を求める要求に 1 つの主サーバーが応答できない場合に、その主サーバーが提供する既存の構成を確認することによって、各サーバーはその他のサーバーの二次サーバーとして動作することができます。各主サーバーは、自動的に 1 つの二次サーバーとして動作します。

DHCP サーバーの IP アドレスの範囲は、そのサーバー上にソフトウェアをインストールして設定する際に指定します。主 DHCP サーバーになると、サーバーは、新規設定を要求するクライアントに対して、自己が受け持つ IP アドレスの範囲から IP アドレスを配付することができます。クライアントが既存の設定の確認を要求した場合は、当該クライアントの IP アドレスを受け持っているサーバーが設定を確認

します。二次サーバーとして動作する場合は、ネットワーク上の別の DHCP サーバーが提供した設定を確認することができます。

二次サービスを提供する場合、DHCP サーバーはネットワーク上の別のサーバーが提供した設定を確認します。これを行うのは、IP アドレスを受け持つ主サーバーが応答することができない場合です。待機時間経過後に、主サーバーに代わって二次サーバーが応答します。

DHCP サーバーを二次サーバーとしてのみ構成することもできます。DHCP サーバーを二次サーバーとしてのみ構成したい場合は、`dhcpconfig` プログラムを使用し、新規設定を要求するクライアントに対して配付する IP アドレスの範囲を持たないサーバーの構成を選択します。この構成では、DHCP サーバーは、データの記憶領域として NIS+ を使用している必要があります。

DHCP サービスは、DHCP が `dhcpconfig` ユーティリティを用いて動作しているマシン上で有効にして構成することができます。このユーティリティを使用すると、起動オプションの設定、DHCP サービスのデータベースの形式と位置の設定、任意のローカル接続またはリモートネットワークの `dhcptab` テーブルと `dhcp_network` テーブルの初期化を行うことができます。

`dhcpconfig` を起動するとメニューが表示されます。このメニューには、DHCP サービスを構成するオプション、BOOTP 中継エージェントを設定するオプション、DHCP の構成または中継サービスの構成を削除するオプション、終了オプションがあります。管理者がメニューオプションのうちの 1 つを選択すると、必要な情報を収集するための一連の質問が表示されます。次に、`dhcpconfig` により、選択した機能をオンに設定するための処理が実行されます。

同じネットワーク上の複数の DHCP サーバーは、NIS+ または NFS を介して DHCP データベースを共有すると、より効率的に動作します。共有を行うと、DHCP サーバーが共通のデータストアを介して通信できるために、冗長性が増えて連携するサービス間で負荷が分散されます。

新規 DHCP クライアントがネットワークに追加されると、そのクライアントは、到達範囲内にある使用可能なすべての DHCP サーバーまたは BOOTP サーバー、あるいはその両方を検出する目的でメッセージを送ります。このメッセージを受け取った DHPC サーバーは最初に、割り当て可能な IP アドレスがあるかどうかを検査します。割り当て可能な IP アドレスがある場合、サーバーは、それがまだ使用されていない IP アドレスであるかどうかを確認します。まだ使用されていない IP アドレスがない場合、サーバーは、その IP アドレスとその他の構成情報をクライアントに提供します。IP アドレスが使用中であった場合、サーバーはその IP アドレスを使用不可とマークし、状態を管理者に通知して、別の IP アドレスを選択します。

クライアントは、独自の条件にもとづいて、クライアント自身に提供された 1 つの IP アドレスを選択し、自己の選択を特定するメッセージを送ります。

サーバーのデータベース

DHCP/BOOTP サーバーは、dhcptab データベースと dhcp_network データベースという 2 種類のデータベースを使用します。

dhcptab データベースは、termcap に似た構文を使用して定義されたマクロを格納しています。この構文を使用すると、ネットワーク管理者はクライアントに戻す DHCP 構成パラメータのグループを定義することができます。現在のところ、77 の定義済みパラメータがあります。

DHCP/BOOTP サーバーは、そのサーバーと同じネットワークに接続されているクライアントによって要求された場合には、ホスト名、ネットワークブロードキャスト通信アドレス、ネットワークのサブネットマスク、IP 最大転送ユニット (MTU) のいずれかを戻します。この情報は、dhcptab 内に明示的に設定する必要はありません。dhtadm コマンドによって dhcptab サーバー構成テーブルを管理します。

分散型 dhcptab テーブルを共有する 2 つのサーバーが存在し、かつ 2 つのサーバーが同じ NIS+ ドメイン内にある場合、管理者はそのテーブル内の DHCP パラメータを設定して 2 つのサーバーに相互のバックアップを行わせることができます。ただし、本来の構成として、それぞれが異なる範囲の IP アドレスを受け持つ必要があります。さらに、クライアントが他のネットワーク上のサーバーに到達することを可能にするため、個々のネットワークに BOOTP 中継エージェントが必要な場合もあります。

dhcp_network データベースは、クライアント識別子と IP アドレスとの間のマップを格納しています。このデータベースの名前は、サポート対象のネットワークにちなんで付けられています。dhcp_network データベースは、DHCP/BOOTP サービスを提供するネットワークごとに 1 つ存在します。dhcp_network データベースは、実行時にサーバーによって動的に検出され、問い合わせを受けます。dhcp_network データベースが存在しないネットワークから受け取ったクライアントの要求は無視されます。

dhcp_network データベースは、IP アドレスとその IP アドレスに関連付けられている構成パラメータに対して、DHCP クライアントのクライアント識別子を対応づけます。このデータベースは、実行時に DHCP サーバーによって検出されますが、検出は DHCP 要求が発信されたネットワークの IP ネットワークアドレスとサブネットマスクを使用して dhcp_network データベース名を生成することにより

行われます。たとえば、10.0.0.0 ネットワークをサポートする dhcp_network データベースは 10_0_0_0 と呼ばれます。dhcp_network データベースは、NIS+ テーブルまたは ASCII ファイルとして存在が可能です。dhcp_network データベースを管理する場合は、pntadm コマンドを使用します。

in.dhcpd デーモンには、DHCP サーバー (オプションの BOOTP 互換モードを含む) および BOOTP 中継エージェントモードという 2 つの動作モードがあります。

BOOTP 中継エージェント

複数のネットワークと、ネットワークを特定するためのネットマスクの使用によって、TCP/IP をベースとしたネットワークの機能が複雑化しています。たとえば、IP を使用したブロードキャスト通信は、ネットワーク同士を接続するゲートウェイを介して行うことはできません。つまり、あるネットワーク上のクライアントは、その他のネットワーク上のサーバーに対して DHCP 要求または BOOTP 要求の一斉同報通信を行うことができません。BOOTP 中継エージェントがゲートウェイを介してサーバーへ要求を送り、次にサーバーからの応答をクライアントに戻す必要があります。

ルーターに組み込み BOOTP 中継エージェントはないけれども、DHCP サーバーにインストールしたサービスの利点をその他のネットワーク内のクライアントにも利用させたいと希望する場合は、それらのネットワーク上に BOOTP 中継エージェントをインストールすることができます。BOOTP 中継エージェントを使用すると、DHCP サーバーが動作していないネットワークから DHCP サーバーへアクセスすることが可能になります。

in.dhcpd デーモンを BOOTP 中継エージェントとして動作させることができます。BOOTP 中継エージェントモードを指定する場合は、オプションの引数により、中継エージェントが BOOTP 要求を転送する必要がある転送先の DHCP サーバーまたは BOOTP サーバーの IP アドレスまたはホスト名のコンマ区切り形式のリストを指定します。このモードでデーモンを開始すると、DHCP データベースがすべて無視されて、デーモンが BOOTP 中継エージェントとして動作します。

BOOTP 中継エージェントは、UDP ポート 68 での受信を待機し、このポートで受信した BOOTP 要求のパケットをコマンド行に指定された転送先へ転送します。中継エージェントは、ローカルなルーターの情報を持つマシン上であれば動作可能であるため、インターネット用のゲートウェイマシンである必要はありません。

–r IPaddr | hostname ... オプションによって、BOOTP 中継エージェントが使用可能になります。netmasks データベースに正しいエントリを作成して、BOOTP 中継エージェントが役割を果たす対象の DHCP サーバーが、外部の BOOTP/DHCP クライアントのネットワークのサブネットマスクを特定できるようにする必要があります。

BOOTP 中継エージェントをインストールした後で、分散型 DHCP データベースにエントリを追加して、BOOTP 中継エージェントを介して要求を送信するクライアントに DHCP サーバーがサービスを提供できるようにする必要があります。

pntadm コマンドのマクロオプション (–M) は、特定の IP アドレスを使用して、クライアントに返す構成パラメータを、ネットワーク管理者が選択できるようにします。このマクロオプションは、サーバーに固有な情報を含むマクロを配信する場合にも使用することができます。この場合は、当該マクロ定義を特定のサーバーが所有するすべての dhcp_network データベースのエントリに含めます。

リース

DHCP/BOOTP サーバーは、dhcptab データベースと dhcp_network データベースに格納されている情報を使用して、クライアントの IP アドレスのリースを計算します。サーバーが調べるのは、dhcptab データベース内の LeaseTim シンボルと LeaseNeg シンボル、選択された dhcp_network データベースレコードの Flags フィールドと Lease フィールドです。

サーバーは最初に、特定された dhcp_network レコードの Flags フィールドを調べます。PERMANENT フラグまたは BOOTP フラグがオンの場合、クライアントのリースは永久的であるとみなされます。

PERMANENT フラグがオンではない場合、サーバーは dhcp_network レコード内の Lease フィールドに表示されているクライアントのリースが期限切れになっているかどうかを検査します。期限切れになっていない場合、サーバーはクライアントが新規リースを要求しているのかどうかを検査します。クライアントの dhcptab パラメータ内に LeaseNeg シンボルが含まれていない場合、クライアントが要求しているリースの延長は無視され、リースは Lease フィールド内に表示されている残り時間に設定されます。

LeaseNeg シンボルが含まれていて、かつ要求されたリースが、現在時刻にクライアントの LeaseTim dhcptab パラメータの値を加えた値以下である場合、サーバーはクライアントのリースをクライアントが要求した値まで延長します。ク

ライアントが要求したリースがポリシーにより許容される値 (ポリシーは LeaseTim の値) より大きい場合は、現在時刻に LeaseTim の値を加えた値に等しいリースがクライアントに与えられます。LeaseTim が設定されていない場合は、デフォルトで LeaseTim の値が 1 時間となります。

第 15 章

DHCP への移行

この章では、DHCP、BOOTP、RARP の各プロトコル間の違いについて説明します。また DHCP の利点と DHCP へ移行する方法についても説明します。

DHCP へ移行する理由

BOOTP または RARP を使用していたユーザーの中には、DHCP との違いや DHCP の利点について疑問に思われる方がいらっしゃるかもしれません。DHCP とそれ以前のプロトコルとの間の主要な違いは、DHCP 以前のプロトコルでは、ホスト情報をサーバーのデータベース内に手動で構成するように設計されていましたが、DHCP では新しく接続されたホストに IP アドレスと構成を動的に割り当てることが可能になりました。

さらに、DHCP のリースのメカニズムにより、IP アドレスの自動的な回収と再割り当ても可能になりました。DHCP は BOOTP のスーパーセットであり、より高度の柔軟性を提供します。DHCP は BOOTP をベースにしており、いくつか機能が追加されていますが、同じプロトコルパケット形式とメカニズムを使用します。DHCP

では、ルーターに組み込み済みの BOOTP 中継エージェントの機能を利用することができ、BOOTP クライアントを直接サポートすることができます。

RARP ではマシンが自己の IP アドレスを見つけ出しますが、DHCP または BOOTP では IP アドレスはクライアントシステムに引き渡されるプロトコルパラメータの 1 つです。RARP の欠点は、その他のパラメータがサポートされていないこと、および RARP を提供するサーバーが、直接に接続されたネットワーク以外にはサービスを提供できないことです。

DHCP と BOOTP のトラフィックは、一般的なルーターに組み込まれている BOOTP 中継エージェントの機能を利用することができます。つまり、ネットワーク管理者は、ネットワークセグメントごとに BOOTP サービスを配置する必要はありません。

手動で設定した IP アドレスをサポートしようとすると、管理者は以下のような複数の困難に直面します。

- IP アドレスがまだ使用中であるかどうかを確実に検出する方法がない。
- ネットワークトポロジを変更する (たとえば、新規のネットワーク IP アドレスを追加する) たびに、管理者が手動で新規 IP アドレスを追加する必要がある。
- あるネットワーク上で構成したホストを別のネットワークへ移動した場合、通信を行うためには移動したホストの構成を手動で変更する必要がある。

DHCP の利点

DHCP サーバーには、これまでの IP アドレスを取得する方法に比べて、優れた点があります。DHCP サーバーが提供できる機能を以下に示します。

1. IP アドレスの重複問題の防止を含む、IP アドレスの自動管理。
2. BOOTP クライアントのサポートが可能、これによりネットワークを BOOTP から DHCP へ容易に移行することが可能。
3. 管理者がリース時間を設定することが可能であり、手動で割り当てた IP アドレスに対しても設定することが可能。
4. 動的な IP アドレスを用いてサービスを提供する対象の MAC アドレスを制限することが可能。
5. 管理者が、BOOTP の場合に可能である範囲を超えた、追加の DHCP オプションの種類を構成することが可能。

6. 動的に割り当てることができる IP アドレスのプール (複数可) を定義することが可能。ユーザーのサーバーで、プールが 1 つのサブネットまたはネットワーク全体になるよう強制するものがありますが、1 つのプールが連続した IP アドレスから構成されることを強制するサーバーは望ましくありません。
7. 別個の IP ネットワーク (またはサブネット) に対する、複数の動的な IP アドレスのプールを関連付けることが可能。この関連付けは、二次ネットワークに対する基本サポートであり、複数の IP ネットワークアドレスまたはサブネット IP アドレスを持つインタフェースの BOOTP 中継としてルーターが動作することを可能にします。

DHCP サーバーの機能の一部ではありませんが、DHCP サーバーを管理する方法に関連したいくつかの機能を以下に示します。

1. 複数のサーバーの集中管理
2. サーバーが動作中で、かつリースが追跡されている状態でも、変更を行う能力。たとえば、IP アドレスをプールに追加したりプールから削除すること、またはパラメータを変更することができます。
3. パラメータに対して広域的変更 (すべてのエントリに対して適用される変更) を行うか、あるいはクライアントまたはプールのグループに対して変更を行う能力
4. リースの監査トレール (たとえば、貸し出し中のリースのログ) の保守

DHCP は、IP アドレスを割り当てる 4 つの方法をサポートします。これらの方法は独立した機能です。特定の 1 つのサーバーに注目すると、そのサーバーはすべての機能を提供できるか、あるいは 1 つも提供できないかのいずれかです。

- 手動。クライアント識別子と IP アドレスとの間の一意な割り当てを管理者が行います。したがって、DHCP サーバーは、この種類の IP アドレスをリースの期限が切れた後でその他のクライアントに再度割り当てることはできません。この種類の IP アドレスの割り当ては、管理者がホストに同じ IP アドレスを保持させ、かつ IP アドレスが使われなくなった際にはそのことを検出したいと希望する場合に便利です。メールのように IP アドレスにより配置が行われるサービスを提供するホストがこの例です。
- 永久。サーバーの管理者が IP アドレスだけを格納したサーバーの構成を作成して、クライアントに提供します。IP アドレスが MAC アドレスと関連付けられた後、サーバーの管理者が介入しない限りその関連付けは永久的です。永久 IP アドレスの割り当てには、割り当てた IP アドレスを自動的に回収することはできないという欠点があります。
- 動的 (期間が制限されたリースを使用します)。サーバーがリースを追跡し、リースの期限が切れて使用可能になると、サーバーが自動的に DHCP クライアント

にそれらの IP アドレスを提供します。管理者による対話は必要ありません。非 BOOTP クライアントに対しては、この種類の IP アドレスを推奨します。

- BOOTP。BOOTP クライアントが使用するために予約されたアドレスです。この方法では、BOOTP クライアント専用の IP アドレスのプールに管理者が入力することが可能です。

移行

DHCP は BOOTP と BOOTP のパケット構造をベースにしているため、大部分のサイトでは DHCP への移行を容易に行うことができます。数多くの DHCP サーバーが、以前の BOOTP クライアントと新しい DHCP クライアントの両方をサポートします。

Solaris DHCP サーバーは、DHCP 照会だけでなく BOOTP 照会も処理するため、BOOTP クライアントは DHCP サーバーからブートすることができます。DHCP クライアントに BOOTP サーバーからの応答を使用するように書き込まれている場合、DHCP クライアントは BOOTP サーバーからブートすることができます。Windows 95 を用いて組み込まれた TCP/IP スタックには、この機能はありません。

サブネット

DHCP クライアントのメッセージは、通常 IP ルーターの機能である BOOTP 中継エージェントによってリモートサーバーへ送信されます。BOOTP 中継エージェントを介して、DHCP サーバーは要求元のサブネットを見分けることができます。BOOTP 中継エージェントは、メッセージの発信元のサブネットを DHCP のメッセージヘッダーに記録します。つまり、DHCP サーバーはその記録を使用して、クライアントが存在するネットワークを判定することができます。

BOOTP サーバーと DHCP サーバーを同じマシン上で動作させることはできません。その理由は、両方のサーバーが同じポート番号を使用するためです。BOOTP 互換モードをオンに設定すると、Solaris DHCP サーバーを BOOTP クライアントとして機能させることができます。

DHCP プロトコルを用いると、すでにリースされた IP アドレスまたは永久 IP アドレスを保持しているクライアントが、別のサブネット上の別の一時リースを取得することができます。この取得は、別の位置へ移動する必要があるマシンにとって役

立ちます。このオプションは、サーバーが当該機能をサポートしている場合に使用可能です。

ルーター

DHCP には不揮発性の記憶領域が必要です。このため、DHCP サービスのタスクはサーバーとは互換性が保たれますが、専用のルーターとは互換性がなくなります。中継用と DHCP 用の両方に構成可能な、サーバーの種類がいくつかあります。たとえば、Web サーバー、ファイアウォールなどの用途で使用できるように設計されたオールインワンのインターネットゲートウェイがあります。ただし、専用のルーターは存在しません。

DHCP の RFC では、DHCP はルーターの構成に使用することを目的としていない旨が明記されています。ルーターの保守および障害追跡においては、構成が自動に設定されるがままにしておくのではなく、正確な構成を把握しておくこと、およびルーターの動作を別のサーバーの動作に依存させないことが重要だからです。

汎用性がより強い特定の種類のコンピュータまたはサーバーを構成して、それらの IP アドレスを DHCP から取得し、ルーターとして動作させることが可能な場合があります。さらに、厳密にはルーターではありませんが、自己のクライアントに与える IP アドレスを DHCP を使用して取得するリモートアクセスサーバーも存在します。

第 16 章

DHCP の管理

この章では、DHCP を管理する方法、すなわち DHCP を実行するネットワークを設定する方法、リース時間ポリシーを決定する方法、BOOTP 中継エージェントを追加する方法について説明します。また、DHCP が使用するデータベースの種類と、特定のデータベース内でマクロを作成する方法についても説明します。さらに、DHCP に実装されたオプション、DHCP に追加可能なオプションについても説明します。

■ 301ページの「主ネットワークインタフェースとして指定する方法」

情報を収集してから DHCP のサービスを設定

DHCP を実行するネットワークを設定する場合は、まず既存のネットワークについての情報を収集する必要があります。必要な情報は、ネットワークトポロジについての情報 (ルーター、スイッチ、その他のネットワークなど) とサービスについての情報 (ネームサービス、ファイルおよび出力サービスなど) です。

リモートネットワーク上のクライアント (すなわち、DHCP サービスを配置する予定のネットワークとは別のネットワーク上のクライアント) をサポートすることを予定している場合は、リモートネットワークのサブネットマスクも収集する必要があります (ただし、リモートネットワークがサブネット化されている場合)。DHCP サービスが使用する `netmasks` テーブルが、この情報を用いて変更済みであることを確認してください。さらに、リモートネットワーク上のルーターの IP アドレスを収集するか、またはルーター検出機能を使用するようにリモートネットワーク上のクライアントを構成する必要があります。

必要な情報をすべて取得したら、ネットワーク内を移動するデータを NIS+ とファイルのどちらに格納するかを決める必要があります。複数サービス環境または事業用の場合は、NIS+ が適しています。単一サーバーまたは小規模な環境の場合は、ファイルが適しています。情報の収集が終了したら、`dhcpconfig(1M)` を実行してリモートネットワークを構成します。

DHCP データ用のデータストアの選択

DHCP ネームサービスの設定では、テーブルを格納する際とホスト情報にアクセスする際に DHCP サーバーが使用するデータストア資源を決定します。`dhcpconfig` スクリプトは、`/etc/default/dhcp` ファイル内に DHCP サービスを設定します。実行時デーモンと管理ユーティリティはこのファイルを使用して、処理の際の問い合わせ先のネームサービスを決定します。

データストアのサービスを選択する方法

まず最初に、dhcpconfig コマンドにより、サーバーが現在使用しているのが NIS+ とファイルのどちらであるのかを判定します。システムが NIS+ を使用中である場合は、nisplus が Enter data store プロンプトにおけるデフォルト値です。システムがファイルを使用中である場合は files がデフォルト値です。

NIS+ を選択して、サーバーが NIS + を実行していない場合は、警告メッセージと NIS+ の設定方法が表示されます。dhcpconfig スクリプトの処理が継続します (ただし、次に DHCP テーブルを作成する際におそらくエラーが発生します)。

複数のサーバーを持つ環境、または事業用の環境の場合は、NIS+ を使用する必要があります。NIS+ を使用すれば、データをサーバー間で共有することができます。単一サーバーのみの場合は、NFS を使用してデータの共有を行う場合を除いて、ファイルを使用することができます。

初期 DHCP テーブルの作成

dhcpconfig スクリプトにより、表 16–1 に示すように、選択したデータストア内に以下の空 DHCP テーブルを作成します。

表 16–1 dhcpconfig スクリプトにより作成するテーブル

dhcptab	DHCP 構成情報テーブル
dhcp_network	DHCP クライアントのマップテーブル、DHCP サーバーのあるネットワークごとに 1 つ

DHCP テーブル

DHCP は、2 種類のデータベース、すなわちネットワークテーブルと dhcptab 構成マクロテーブルとを使用します。これらのデータベースは、NIS+ を使用している場合は NIS+ テーブルであり、NIS+ を使用していない場合はファイルです。

DHCP ネットワークテーブル

DHCP ネットワークテーブルは、IP アドレスの割り当てに関連する情報を格納しています。ネットワークごとに別個のネットワークテーブルがあります。DHCP において dhcp_network テーブルと呼ばれるテーブルの名前は、サービスを提供しているネットワークの IP アドレスから派生しています。たとえば、ネットワーク 120.146.5.0 のネットワークテーブルは、IP アドレス指定の中のピリオドを下線に置換して 120_146_5_0 となります。

DHCP 内の各サブネットには、サブネット内のクライアントのエントリを格納している dhcp_network テーブルがあります。ブートしたクライアントからのパラメータを求める要求に DHCP サーバーが応答すると、そのクライアントの dhcp_network エントリとして情報が記録されます。このテーブルには、クライアントの IP アドレスと、dhcptab テーブルへのポインタとが含まれています。

ネットワークテーブルは、以下の固有情報を格納しています。

- IP アドレス。割り当て済みと未割り当ての両方。
- クライアント識別子 (割り当て済みレコードに対してのみ)
- リースの有効期限
- リースの種類 (動的、永久、手動、使用不可、BOOTP のみのいずれか) を表すフラグ
- IP アドレスごとの dhcptab 構成マクロの名前
- オリジナルのクライアント IP アドレスを所有するサーバーの IP アドレス

ネットワークテーブルは、特定のネットワークに対して DHCP サーバーが割り当てることができる IP アドレスのリストとして機能します。各ネットワークには独自のネットワークテーブルがあります。ネットワークテーブルの基本要素は IP アドレスのリストです。テーブル内のその他の要素は、すべて IP アドレスとの関係において意味を持ちます。たとえば、クライアント ID は特定の IP アドレスが現在割り当てられているクライアントを特定します。IP アドレスが未割り当てである場合、その IP アドレスのクライアント ID は 0 です。有効期限も 0 です。IP アドレスが割り当て済みである場合は、クライアント ID とリースの有効期限が記入されています。

特定の実装状態では、クライアント ID はネットワークの種類を表す接頭辞を付けてクライアントマシンのハードウェアアドレスになります。たとえば、イーサネットアドレスを持つクライアントのクライアント ID が 0102608BA614C1 である場合は、01 によりクライアントがイーサネットネットワークであることが示されます。DHCP の実装状態によっては、その他の識別子 (DNS 名やプロパティ番号など) を

使用する場合があります。重要なことは、クライアント ID はネットワーク内で一意である必要があることです。

IP アドレスが割り当て済みの場合、その IP アドレスのリースの有効期限は特定の日付と時刻に設定されるか、または “No Expiration” とマークされます。

lease フラグと dhcptab 構成マクロの名前は、IP アドレスがクライアントに割り当てられているかどうかにかかわらず同じです。クライアントが特定の IP アドレスを取得すると、lease フラグにより指定されたリースの種類と、プロパティ名により指定された構成も取得します。lease フラグは、IP アドレスを割り当てることができる条件を表示します。pntadm コマンドにより、dhcp_network テーブルを管理することができます。例 16–1 に pntadm の出力例を示します。

例 16–1 pntadm –P *129.146.86.0* の出力例

```
Client ID       Flags  Client IP        Server IP       Lease          Macro     Comment
                                                        Explanation

010800207CBA2C   04    129.146.86.153   129.146.86.181   Zero          mrcoffee
0108002022519C   00    129.146.86.205   129.146.86.181   7/3/1996      inet11
01080011043B65   08    129.146.86.29    129.146.86.181   Zero          inet11
0100A024A9BCEE   08    129.146.86.198   129.146.86.181   7/22/1996     inet11
0100A024A791DE   00    129.146.86.200   129.146.86.181   8/4/1996      inet11
0100A02463D6EC   00    129.146.86.199   129.146.86.181   8/1/1996      inet11
0100A024636AB7   00    129.146.86.201   129.146.86.181   8/3/1996      inet11
010080C72EE4A3   00    129.146.86.206   129.146.86.181   7/5/1996      inet11
010020AF4A3B31   0     129.146.86.214   129.146.86.181   Zero          hobbs
00               00    129.146.86.202   129.146.86.181   Zero          inet11
```

dhcptab 構成テーブル

dhcptab テーブルは、クライアントの構成に関連する情報を格納しています。このテーブルは、ネットワーククライアントを構成するのに必要な全情報を格納する、一連のマクロ定義として編成されます。クライアントは、ネットワークテーブルから IP アドレスを割り当てられる際に構成を取得します。IP アドレスに関連付けられたマクロ名は、dhcptab テーブル内のマクロ名に対応します。クライアントは、ネットワークテーブルから IP アドレスを取得した後に、dhcptab テーブルからネットワーク構成を取得します。

DHCP サーバーの初期構成の際に、構成済みネットワークごとに dhcptab テーブルとマクロが作成されます。各マクロには、ネットワークに固有の情報、すなわちサブネットマスク、ネットワークブロードキャスト通信アドレス、IP パケット生存時間、データグラムの最大サイズ、デフォルトのルーター、静的送信経路、DNS ドメイン、NIS ドメイン、DNS サーバー、NIS サーバーのうち、サーバーの構成時に使用可能なものが格納されます。

マクロ内に格納されている情報を変更することによって、クライアントマシンがネットワークを利用する方法を制御することができます。たとえば、特定のクライアントマシンが使用するマクロの名前を変更すると、そのマシンのネットワーク構成が変更されます。別の例としては、あるマクロ内の 1 つのオプションを変更することにより、そのマクロセットを使用する全マシンの動作が変更されます。IP アドレスを管理する能力は、DHCP の主要機能の 1 つです。dhtadm コマンドにより、dhcptab サーバー構成テーブルを管理します。例 16–2 にdhtadm の出力例を示します。

例 16–2 dhtadm –P の出力例

```
Name            Type    Value

mrcoffee        Macro   :Subnet=255.255.255.0:Router=129.146.86.1:Broadcst=129.146.86.255: \
                        :BootSrvA=129.146.86.175:BootFile="/export/root/JavaDesktop/kona": \
                        :NISservs=129.146.86.33:NISdmain=sunsoft.eng.sun.com: \
                        :DNSdmain=Eng.Sun.COM: \
                        :DNSserv=129.146.1.151 129.146.1.152 129.144.1.57 129.144.134.19: \
                        :Include=Locale: \
                        :Timeserv=129.144.1.3:LeaseTim=3600:T1Time=1800: \
                        :T2Time=3060:

Locale          Macro   :UTCoffst=25200:SN_TZ="PST8PDT":

inet11          Macro   :Include=Locale:Timeserv=129.146.86.181:LeaseTim=259200: \
                        :DNSdmain=Eng.Sun.COM: \
                        :DNSserv=129.146.1.151 129.146.1.152 129.144.1.57 129.144.134.19:

hobbs           Macro   :Subnet=255.255.255.0:Router=129.146.86.1:Broadcst=129.146.86.255: \
                        :BootSrvA=129.146.86.32:BootFile="819256D6.PREP":

129.146.89.0    Macro   :Subnet=255.255.255.0:Router=129.146.89.1:Broadcst=129.146.89.255: \
                        :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.89.33: \
                        :NetBNms=129.146.171.31:NetBNdT=8:

129.146.88.0    Macro   :Subnet=255.255.255.0:Router=129.146.88.1:Broadcst=129.146.88.255: \
                        :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.88.33: \
                        :NetBNms=129.146.171.31:NetBNdT=8:

129.146.87.0    Macro   :Subnet=255.255.255.0:Router=129.146.87.1:Broadcst=129.146.87.255: \
                        :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.87.33:
                        :NetBNms=129.146.171.31:NetBNdT=8:

129.146.86.0    Macro   :Broadcst=129.146.86.255:Subnet=255.255.255.0:MTU=1500: \
                        :Router=129.146.86.1:NISdmain=sunsoft.eng.sun.com: \
                        :NISservs=129.146.86.33:NetBNms=129.146.171.31:NetBNdT=8: \
                        :BootSrvA=129.146.86.32:

129.146.85.0    Macro   :Subnet=255.255.255.0:Router=129.146.85.1:Broadcst=129.146.85.255: \
                        :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.85.33: \
                        :NetBNms=129.146.171.31:NetBNdT=8:

129.146.84.0    Macro   :Subnet=255.255.255.0:Router=129.146.84.1:Broadcst=129.146.84.255: \
                        :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.84.33: \
                        :NetBNms=129.146.171.31:NetBNdT=8:
```

```
129.146.83.0  Macro   :Subnet=255.255.255.0:Router=129.146.83.1:Broadcst=129.146.83.255: \
                      :NISdmain=sunsoft.eng.sun.com: \
                      :NISservs=129.146.83.33:NetBNms=129.146.171.31:NetBNdT=8:

129.146.82.0  Macro   :Subnet=255.255.255.0:Router=129.146.82.1:Broadcst=129.146.82.255: \
                      :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.82.33: \
                      :NetBNms=129.146.171.31:NetBNdT=8:

129.146.81.0  Macro   :Subnet=255.255.255.0:Router=129.146.81.1:Broadcst=129.146.81.255: \
                      :NISdmain=sunsoft.eng.sun.com:NISservs=129.146.81.33: \
                      :NetBNms=129.146.171.31:NetBNdT=8:

SN_TZ         Symbol  Vendor=SUNW,13,ASCII,1,0
```

DHCP の各サブネットの構成

この節では、dhcpconfig を使用し、各サブネットについての以下の 3 つの質問に対する答えをもとにして、サブネットを構成する方法について説明します。

- 使用されていない IP アドレスを割り当てる場合に、検索を希望する IP アドレスの範囲はどこか。
- 特定のサブネットに対する要求に DHCP デーモンが応答することを希望するか。
- 監視された各サブネットについて、動的に割り当てられた IP アドレスの割り当てを希望するクライアントの数はいくつか。

DHCP の各サブネットを構成する方法

dhcpconfig スクリプトにより、サーバーシステム上に構成するサブネットごとに、dhcp_network テーブルと呼ばれるテーブルを作成します。テーブル名は IP アドレスと同じですが、小数点は下線に置換されます。たとえば、サブネット 129.148.5.0 の dhcp_network テーブルは、DHCP が使用しているネームサービス内では 129_148_5_0 です。これは、NIS+ の場合は org_dir オブジェクト内のテーブルであり、ファイルの場合は /var/dhcp ディレクトリ内のファイルです。

DHCP が管理しているクライアントシステムごとに、dhcp_network テーブル (クライアントマシンが接続されているサブネットに対応するテーブル) 内にエントリが 1 つあります。エントリが永久である場合もありますが、この場合は IP アドレスが永久的にマシンに割り当てられています。あるいはエントリが動的である場合もありますが、この場合はクライアントが最初に構成される際に DHCP サーバーが IP

アドレスを割り当てて、さらに IP アドレスを使用できる時間の長さを指定したリースを与えます。この段階で設定するのは、これらの動的クライアントです。永久クライアントは、DHCP の環境をすべて構成した後で、pntadm を用いて設定することができます。

DHCP サービスデーモンの開始

この節では、dhcpconfig スクリプトが実行する、以下の 3 つの機能について説明します。

- DHCP デーモンプロセスの start/stop スクリプトを、/etc/init.d ディレクトリへインストールする
- このスクリプトへのリンクを、/etc/rc{0,3}.d ディレクトリ内に設定する
- デーモンプロセスを開始する

start/stop スクリプトの名前は dhcp であり、リンクは S34dhcp (デーモンを開始する場合) と K34dhcp (デーモンを停止する場合) です。このスクリプトは、デーモンプロセス実行用の標準 SVR4 手続きをブート時に実行します。

デーモンプロセス in.dhcpd を開始します。in.dhcpd デーモンは DHCP サーバープロセスであり、クライアントの要求に応答して、dhcptab テーブル内に確立済みのネットワーク構成を転送します。

リース時間ポリシー

DHCP は IP アドレスを動的に割り当てるメカニズムを提供します。IP アドレスにはリース期間が伴っており、永久または一時に設定することができます。ユーザーサイトのポリシーとして、一時 IP アドレスと永久 IP アドレスの数を決定し、さらに一時 IP アドレスのリース期間を決定する必要があります。

DHCP サービスを最大限利用するには、DHCP に IP アドレスの割り当てを動的に管理させることが最善です。DHCP を用いた場合には、クライアントとサーバーが特定の期間に渡る IP アドレス設定の貸し出し (リース時間と呼ぶ) についてネゴシエーションを行います。dhcptab 内の特定のマクロ定義の LeaseTim シンボルと LeaseNeg シンボルを使用することによって、サーバー、ネットワーク、クライアントのベンダークラス、個別のクライアント IP アドレスのいずれかをベースにしてリース時間ポリシーを設定することができます。

LeaseNeg シンボルと LeaseTim シンボルを使用すると、ユーザーサイトのポリシーを設定することができます。LeaseTim は相対的な時間であり、24時間、2 時間、または 10 時間などとなります。クライアントに IP アドレスが割り当てられると (または、すでに割り当てられている IP アドレスのリースについて再度ネゴシエーションを行うと)、クライアントが DHCP の回答を受け取った絶対時間に LeaseTim の値が追加されます。絶対時間は現在時刻であり、たとえば 1996 年 9 月 27 日です。絶対的な現在時刻に LeaseTim の値を加算した時刻は、IP アドレスについてのクライアントのリースが期限切れとなる絶対時間として、クライアントの dhcp_network レコード内に格納されます。

LeaseTim シンボルの設定は、クライアントに対して許可することができる、リースの最大時間間隔を定義します。一般的にはこの値を相対的に小さな値にしておく必要がありますが、その理由は、クライアントとサーバーの同期を維持するため、使用されていない IP アドレスを遅滞なく回収するため、さらにネットワークの再番号付けをより円滑に行うためです。

ただし、DHCP サーバーが使用不可能になった場合に DHCP サービスを実行しているマシン (複数可) が修復されるまでの間、クライアントが機能を継続することができる程度には大きい値である必要があります。1 ～ 3 日が最善の LeaseTim ポリシーとなります。所定の環境において適切に動作する値を選択してください。

LeaseNeg シンボルは、リースの期限が切れる前にクライアントがサーバーとリースについて再度ネゴシエーションを行うことを可能とするかどうかを決定します。このシンボルが存在する場合、クライアントはリースについて再度ネゴシエーションを行うことができます。LeaseNeg が存在する場合、クライアントは既存の接続に対してリース関連の割り込みを行わないでネットワーク上で動作することができます。

IP アドレスよりマシンの方が数が多く、したがって IP アドレスの使用に時間制限を実施したい環境では、LeaseNeg を省略する方が実用的です。このような場合の例として、コンピュータサイエンスの教室内のマシンに対して時間制限を実施する場合があります。LeaseTim と同様に、LeaseNeg を dhcptab の各種マクロ内で使用することができます。詳細は、dhcptab(4) および dhcp_network(4) を参照してください。

サービス (たとえば、メールや Web ページ) をエクスポートするマシンは IP アドレスを保持する必要がありますが、当該ノードが使用する IP アドレスがもはや使用されていない場合に (おそらくは廃棄された場合に)、その事実を検出可能であることを希望する場合もあります。この希望は、手動で割り当てられた旨を (必要に応じて割り当て者も) 当該ノードの dhcp_network レコードにマークし、そのノードのフラグフィールドを MANUAL に設定することによって実現することができます。た

とえば、ホスト名が gandalf でネットワークが 10.50.0.0 である場合は、**pntadm gandalf -f MANUAL 10.50.0.0** と入力します。

永久リースで IP アドレスを割り当てることもできますが、DHCP サービスを使用して自動的に IP アドレスを回収することはできません。したがって、永久 IP アドレスの数は最小限の管理可能な数に抑えてください。

DHCP サービスは、このフラグフィールドを使用して、dhcp_network レコードエントリの期限が切れて回収可能となる時点を判定します。この値は、pntadm コマンドの e オプションを使用することによって変更できます。このコマンドを使用すると、クライアントのリースの有効期限を過去の時間に設定することができますが、クライアントとそのクライアントのユーザー (複数可) に悪影響を及ぼすことを避けるために、未来の時間にのみ設定してください。

DHCP サービスが動的に IP アドレスを割り当てるか、または既存の有効期限について再度ネゴシエーションを行うたびに、dhcp_network テーブル内のこのフィールドが更新されます。

lease フラグは IP アドレスを割り当てる条件を表します。フラグの設定は、以下を結合したものになります。

0 (Dynamic)	IP アドレスのこのリースには有効期限があります。リースが期限切れとなった場合は、更新が可能である旨がサイトのポリシーに指示してあれば更新することができます。現在のクライアントがリースを更新しない場合、その IP アドレスは別のクライアントに割り当てることができます。フラグが 0 に設定されている場合は、リース時間を変更することができます。
1 (Permanent)	この IP アドレスのリースは永久的に割り当てられており、クライアントがリース時間を変更することはできません。ただし、IP アドレスを使用しているクライアントがその IP アドレスの割り当てを解除することは可能です。割り当て解除された IP アドレスは、別のクライアントに割り当てることができます。
2 (Manual)	この IP アドレスは特定のクライアントのマシンに割り当てられます。クライアントが割り当てを解除することはできません。フラグが 2 に設定されている限り、その IP アドレスを再度割り当てることができるのは、管理者が手動で変更した場合だけです。

4 (Unusable)　この IP アドレスは使用できません。フラグを 4 に設定することにより、IP アドレスの割り当てを防止することができます。DHCP サーバーは、IP アドレスの配置を試行してそれがすでに使用中であることがわかった場合に、その IP アドレスを使用不可とマークします。DHCP サーバーは、IP アドレスを割り当てる前に、通常は ping コマンドを用いてすでに使用中であるかどうかを確認します。この設定は、dhcpconfig 内に構成可能です。

フラグが設定の結合であることがあります。たとえば、フラグが 3 に設定されている場合は、1 と 2 を結合したものです。つまり、この設定は永久かつ手動という設定であり、この IP アドレスのリースは永久リースで、かつ管理者が割り当てたリースになります。

BOOTP 中継エージェントの設定

最初に、組み込み中継エージェントがルーター (単数の場合も複数の場合もあります) にあるかどうかを判定します。組み込み中継エージェントがルーターにある場合は、マニュアルを読んで中継エージェントの使用法を理解してください。組み込み中継エージェントがルーターにない場合は、クライアントネットワーク上にある、中継エージェントとして機能させる Solaris マシンを選択します。マシンに SUNWdhcsr と SUNWdhcsu をインストールしてから、dhcpconfig を実行して Configure BOOTP relay agent を選択します。

希望の BOOTP/DHCP 要求送信先の BOOTP/DHCP サーバーの IP アドレス、またはホスト名を入力します。

標準 DHCP オプション

Solaris DHCP サーバーでは、標準 DHCP オプションがすべて導入されます。これらのオプションには、以下のネットワーク情報が含まれています。

- サーバー名
- ルーターのアドレス
- DNS ドメイン名

- TCP 生存時間
- 接続オプション
- サーバーオプション
- IP 層オプション
- 時間オプション

ベンダーオプション

ベンダーオプションは、DHCP クライアントソフトウェアのベンダーが定義する DHCP オプションです。クライアントは、構成を求める要求を送る際にベンダーのクライアントクラスを組み込みます。`dhcptab` データベース内にこのクライアントクラスと一致するクライアントクラスがある場合には、そのクラスに対して指定されているオプションとその他の構成オプションがクライアントに送信されます。Solaris DHCP サーバーを構成して、任意の DHCP クライアントベンダーのオプションをサポートすることができます。

ベンダーオプションとサイトオプションの追加

追加のベンダーオプションまたはサイトオプションを作成するには、以下を定義する必要があります。

- ベンダークラス。これは、ベンダーによってクライアントを識別するのに使用する名前です。クラス名は ASCII 文字列です。サイトオプションを定義する場合は指定しません。
- 値の型。これは、当該オプションに格納されるデータの型を指定します。サポートされているデータ型は以下のとおりです。
  - ASCII テキスト
  - オクテット。2 進データの ASCII 表現
  - IP。インターネットアドレスのドット区切り 10 進表現
  - 数値。8 ビット、16 ビット、32 ビット、または 64 ビットの数
- オプションコード。これは、新規オプションに割り当てる DHCP オプション番号です。ベンダーコードは 1 と 254 の間、サイトコードは 128 と 254 の間が可能です。

- 粒度。これは、当該オプションの単一インスタンスを構成する、当該値型であるオブジェクトの数を指定します。たとえば、静的送信経路シンボルは送信経路のリストですが、各送信経路は 2 つの IP アドレスから構成されるので、値の型は *IP* と定義し、粒度は *2* と定義します。
- 長さ。これは、当該オプションにおける最小許容粒度を指定します。たとえば、サブネットマスクは 1 つの IP アドレスだけでもかまわないので、サブネットマスクのオプションの長さは 1 となります。0 という値は、項目の数が可変である (最大 16) ことが許容されることを表します。

サイトオプションはサイトに固有であるため、必要な任意のオプションを作成できますが、ベンダーオプションの場合は、特定のクライアントベンダーに対して必要なオプションだけ作成できます。オプションは定義済みのものもありますが、作成する必要があるものもあります。作成する場合には、特定のベンダーに適用するベンダーオプションのリストをサーバー上に作成することが必要な場合があります。リストの例はクライアントのベンダーが提供します。

マクロ定義の作成

`dhcptab` テーブルのマクロを作成する場合は、関連する標準オプション、ベンダーオプション、サイトオプションをすべて指定する必要があります。使用可能なオプションをすべて指定する必要はありません。指定する数は、ネットワークの構成に応じて異なります。

IP アドレスのリース

IP アドレスのリースは、デフォルトでは一時として割り当てられます。一時リースは、ユーザーとユーザーのマシンが頻繁にサブネットを変更する場合、またはシステムへの出入りが激しい場合に便利です。

サイトごとに、当該サイトでの一時リースを更新可能とするかどうかを指定することができます。このサイトのポリシーは、`LeaseNeg` シンボルを用いてプロパティテーブル内に設定することができます。このシンボルを省略した場合は、リースが期限切れとなった際にクライアントがリースについて再度ネゴシエーションを行うことはできません。IP アドレスが期限切れとなった際にクライアントがその IP アドレスを更新しない場合、その IP アドレスは再使用することができます。

カスタマイズ例

ネットワーク 126.147.100.0 の NIS サーバーの値を変更する場合は、以下のようにします。

1. マクロ `129.147.100.0` を以下のように編集します。

```
dhtadm -M -m 129.147.100.0 -e 'NISserv = 129.147.100.1 129.147.100.2'
```

2. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

3. 以下のように入力します。

```
/etc/init.d/dhcp start
```

あるいは、2 と 3 の代わりに、`in.dhcpd` に対して `–t` オプションを使用します。

10-15 のアドレスを 129.147.100.0 へ追加するには、以下のようにします。

```
for addr in 10 11 12 13 14 15
do
 pntadm -A 129.147.100.$addr -m server -h hundred-$addr 129.147.100.0
done
```

タイムゾーン SN_TZ のシンボル定義を追加するには、以下のようにします。

1. 以下のように入力します。

```
dhtadm -A -s SN_TZ -d  'Vendor="SUNW.PCW.LAN SUNW.Solaris", 13, ASCII, 1, 0'
```

2. 以下のように入力します。

```
dhtadm -M -m Locale -e ':SN_TZ = "EST5EDT4":'
```

3. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

4. 以下のように入力します。

```
/etc/init.d/dhcp start
```

あるいは、–t オプションを使用します。

jurassic マクロから Timeserv 値を削除するには、以下のようにします。

1. 以下のように入力します。

```
dhtadm -D -m jurassic -e 'Timeserv='
```

2. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

3. 以下のように入力します。

```
/etc/init.d/dhcp start
```

あるいは、–t オプションを使用します。

常にホスト名をサーバー jurassic のクライアントに戻すには、以下のようにします。

1. 以下のように入力します。

```
dhtadm -M -m jurassic -e 'Hostname= _NULL_VALUE_'
```

2. 以下のように入力します。

```
/etc/initd/dhcp stop
```

3. 以下のように入力します。

```
/etc/init.d/dhcp start
```

あるいは、-t オプションを使用します。

canoepoint という名前のホストと 1 つの IP アドレスとの間の、ホスト名 ~ IP アドレス間関連付けを維持することが重要な場合は、peds ネットワーク上の canoepoint エントリを MANUAL とマークします。

1. 以下のように入力します。

```
pntadm -M canoepoint -f MANUAL peds
```

あるいは

2. 以下のように入力します。

```
pntadm -M canoepoint -f 02 peds
```

129.147.100.87 を BOOTP かつ永久とマークするには、以下のようにします。

1. 以下のように入力します。

```
pntadm -M 129.147.100.87  -f 'BOOTP + PERMANENT' 129.147.100.0
```

あるいは

2. 以下のように入力します。

```
pntadm -M 129.147.100.87 -f 09 129.147.100.0
```

保守

このシェルスクリプトは最初に、使用不可とマークされている IP アドレスをすべて検査して、使用されていないかどうかを確認します。使用されていない場合は、このスクリプトが IP アドレスを回収します。

```
#!/bin/ksh
# This shell script reclaims addresses which were marked as unusable, after
# first verifying that they're no longer in use.

if [ ${#} -eq 0 ]
then
     echo "reclaim <network> ..." >&2
     exit 1
fi

while [ ${#} -ne 0 ]
do
     pntadm -P ${1} | awk ' $2 == 04 { printf("%s %s\n", $1, $3); }' |
     while read cid addr
     do
        if [ ${?} -ne 0 ]
        then
             pntadm -M ${addr} -i 00 -f DYNAMIC -e 0 ${1}
             if [ ${?} -eq 0 ]
             then
                 echo "${addr} has been reclaimed from client ${cid}."
             fi
             else
                 echo "${addr} is in use!" >&2
             fi
     done
     shift
done
exit 0
```

Solaris DHCP クライアントを有効にする方法

デフォルトでは、Solaris DHCP クライアントは無効にされています。有効にするには、DHCP を用いて構成したいと希望するネットワークインタフェースごとに、DHCP イネーブルファイルを 1 つ作成する必要があります。DHCP イネーブルファイルの書式は `/etc/dhcp.`*interface_name* であり、*interface_name* は DHCP によって構成したいと希望するネットワークインタフェースの名前です。

たとえば、DHCP を使用してネットワークインタフェース `le1` を構成したい場合は、空ファイル `/etc/dhcp.le1` を作成します。DHCP を使用して構成したいネットワークインタフェースが複数ある場合は、インタフェースごとに DHCP イネーブルファイルを 1 つ作成する必要があります。

ブートプロセスが一時停止する時間の増加

DHCP を使用してインタフェースを構成すると、ブート時間が増加することがあります。特に、クライアントの要求に応答する DHCP サーバーが存在しない場合には、インタフェースごとに約 30 秒の遅延が発生します。ネットワークインタフェースが構成されるまで、Solaris DHCP クライアントがブートプロセスを一時停止する (所要時間の長さにかかわらず) ことを希望する場合は、ネットワークインタフェースの DHCP イネーブルファイル (`/etc/dhcp.`*interface_name*) を編集して、`wait forever` というフレーズを追加します。

クライアントがブートプロセスを一時停止する時間をより短くしたいと希望する場合は、キーワード `forever` を使用する代わりに、待機する秒数を指定することができます。たとえば、DHCP がネットワークインタフェースを構成する時間として 1 時間待機してからブートプロセスを継続したいと希望する場合は、`wait 3600` と指定します。

注 - 一時停止時間が経過した場合でも、Solaris DHCP クライアントは、ネットワークインタフェースの構成が正常終了するまで非同期的に構成を継続します。この継続を回避するために、`ifconfig(1M)` コマンドに `drop` オプションを指定することができます。たとえば、`ifconfig` *le0* `dhcp drop` とします。これにより、指定されたインタフェース (この例では *le0*) が DHCP エージェントの制御から削除されて、非同期的なアドレス割り当て試行が終了します。

主ネットワークインタフェースとして指定する方法

大部分の DHCP 構成パラメータは 1 つのネットワークインタフェースに固有ではありません。パラメータには、より一般的な情報を指定します。この種の一般パラメータの例として、NIS サーバー、NIS ドメイン、DNS サーバー、DNS ドメインがあります。Solaris マシンに 1 つのネットワークインタフェースだけがある場合は、一般パラメータとインターフェースに固有なパラメータとを区別する必要はありません。

マシンに複数のネットワークインタフェースがあり (すなわち、複数のホームがあり)、DHCP が複数のインタフェースの構成を行う場合は、複数のセットの一般構成パラメータを受け取って、パラメータ同士が衝突する可能性があります。たとえば、DHCP を使用してインタフェース *le0* を構成する際に受け取った DNS パラメータを使用すべきなのでしょうか、それとも *le1* 用に受け取った DNS パラメータを使用すべきなのでしょうか。

1 つのネットワークインタフェースを主ネットワークインタフェースとして指定すれば、Solaris DHCP クライアントはこの問題を解決することができます。インタフェースに固有なパラメータ (たとえば、サブネットマスク) は各インタフェースから取り出され、一般パラメータは、主インタフェースから受け取った DHCP 情報だけから取り出されます。

ネットワークインタフェースを主インタフェースとして指定するには、そのインタフェースの DHCP イネーブルファイルにキーワード `primary` を追加します。たとえば、*qe2* を主インタフェースとして使用したいと希望する場合は、`/etc/dhcp.qe2` を編集して、`primary` という語を追加します。

キーワード `primary` が追加されていない場合 (主インタフェースとして指定されているインタフェースがない場合)、Solaris マシンは、構成が最初に正常終了したインタフェースからパラメータを受け取ります。

DHCP/BOOTP の有効利用を制限するネットワークトポロジ

DHCP クライアントと BOOTP クライアントは最初、ローカル IP ネットワークについての情報を持っていません。したがって、自己の IP アドレスとして 0.0.0.0

(デフォルトのネットワークアドレス) を使用します。DHCP 要求または BOOTP 要求が、これらのクライアントから 255.255.255.255 IP アドレス (ブロードキャスト通信アドレス) へ送信され、ローカル IP ネットワークに接続されている全 IP デバイスが受信します。

DHCP サーバーと BOOTP サーバーは、以下の要素をベースにしてクライアントの IP ネットワークのアタッチメントを判定します。

1. DHCP 要求または BOOTP 要求を受け取ったネットワークのハードウェアインタフェース
2. 受け取った DHCP 要求または BOOTP 要求は、BOOTP 中継エージェントからのものであったかどうか

   BOOTP 中継エージェントは、DHCP クライアントまたは BOOTP クライアントと同じ IP ネットワークに接続されている、自己のネットワークのハードウェアインタフェースの IP アドレスを挿入します。この IP アドレスが欠落している場合は、クライアントが直接接続された IP ネットワーク上にある旨の信号がサーバーへ送信されます。この IP アドレスが存在する場合は、サーバーから離れたリモート IP ネットワークにクライアントが接続されていること、および BOOTP 中継エージェントの IP アドレスを使用してサーバーがクライアントへ応答を返信することを表しています。
3. クライアントが接続されている IP ネットワークがサブネット化されているかどうか

   サーバーは、IP アドレスをキーとして使用して、`netmasks` テーブル (サブネットのマスク情報を格納しています) の内容を調べます。この場合に使用する IP アドレスは以下のうちのいずれかです。

   - クライアントが直接接続された IP ネットワーク上にある (パケットの中継アドレスフィールド内の 0.0.0.0 という IP アドレスによって示されます) 場合は、サーバーのネットワークのハードウェアインタフェースの IP アドレス
   - BOOTP 中継エージェントがクライアントの要求内に IP アドレスを指定していた場合は、その指定された IP アドレス

クライアントの IP ネットワークのアタッチメントを判定するこの手順が有効なのは、ネットワークのハードウェア媒体 (たとえば、イーサネット) 上に存在する IP ネットワークが 1 つだけである場合のみです。複数のネットワークハードウェアインタフェースを使用するか、または複数の論理インタフェースを使用することによって、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 IP ネットワーク環境では、DHCP は適切に動作しません。この場合には、DHCP クライアントの要求が全ネットワークハードウェアインタフェース上に表示され、「外

観上は」そのクライアントがすべての IP ネットワークに同時に接続されているかのように見えます。DHCP サーバーは IP アドレスを動的に要求元のクライアントへ割り当てるため、そのクライアントへ割り当てるべき IP アドレスをサーバーが決定することはできません。それは、その時点でサーバーが保持している IP アドレスの妥当性検査を試行すると、DHCP クライアントは、割り当てられたネットワーク上だけではなく、すべての論理 IP ネットワーク上に存在するように見えるからです。

このようなネットワークトポロジは回避する必要があります。そのためには、より効率的なサブネット化を行い、可変長サブネットマスク (VLSM) を使用して IP ネットワーク間のハードウェア媒体のマップを一対一に保持するか、あるいは、ただ 1 つの論理 IP ネットワークがサービスの対象となるように DHCP または BOOTP のサービスを構成します。詳細は、`netmasks(4)` を参照してください。

第 17 章

DHCP の障害追跡

この章では、DHCP の使用時に検出される可能性がある問題の障害追跡を行う方法について説明します。最初の DHCP サーバーをインストールして構成する時点で発生する可能性のある問題に対する解決策も説明します。DHCP サーバーの構成スクリプト (dhcpconfig) についての内容説明 (種々のスクリプト構成要素の目的と、スクリプトがインストール手順を実行する過程) も記載されています。さらに、ネットワークに DHCP クライアントを初めて追加する際と後続して追加する際に検出される可能性がある問題についても説明します。

- 324ページの「クライアントがDHCP サーバーに接続できない場合」
- 327ページの「エラーメッセージを調べる」
- 331ページの「BOOTP 互換モードにおいて、一部のクライアントが DHCP サーバーからブートしない場合」
- 331ページの「NIS + 構成の問題の診断」
- 333ページの「ネームサービス構成の問題の診断」
- 335ページの「マクロの変更がクライアントに伝達されない場合」

方法および注意事項

以下の障害追跡手法は、原因を特定できない場合の問題解決に役立ちます。

- snoop コマンドを使用して、ネットワークのトラフィックを監視します。
- DHCP クライアントをデバッグモードで動作させます。
- DHCP サーバーをデバッグモードで動作させます。
- DHCP クライアントをリブートします。
- DHCP サーバーを停止します。その後で再度開始します。

この章では、上記の手法を詳細に説明します。また、本書を使用しても問題を解決することができない場合の問い合わせ先を紹介します。

snoop を使用してネットワークのトラフィックを監視

snoop コマンドを使用してネットワークのトラフィックを監視することができます。

▼ snoop を使用してネットワークのトラフィックを監視するには

1. クライアントと同じサブネット上の **Solaris** サーバーまたは **BOOTP/DHCP** 中継エージェントに、スーパーユーザーとしてログインします。

2. snoop コマンドを使用してネットワークのトラフィックを監視します。たとえば、以下のように入力します。

```
snoop -o /tmp/output udp port 67 または udp port 68
```

3. クライアントをブートして、クライアントとサーバー (複数可) との間の **DHCP** メッセージの交換を監視します。

4. 以下のように入力します。

```
snoop -i /tmp/output -x 0 -v
```

クライアントのハードウェアアドレスを指定することによって snoop の適用範囲を制限することができます。DHCP/BOOTP プロトコルを解釈できる snoop は、Solaris 2.5 オペレーティング環境およびその互換バージョンで使用できます。

DHCP クライアントをデバッグモードで動作

DHCP クライアントをデバッグモードで動作させると、クライアントとサーバーとの間で進行中のほとんどの対話が明らかになります。クライアントが動作しているベースの製品については、関連のマニュアルを参照してください。

▼ Solaris クライアントをデバッグモードで動作させるには

DHCP クライアントのデバッグは、DHCP クライアントをブートした後でのみ可能です。DHCP に問題が発生した場合は、DHCP を無効にしてブートする必要があります。以下の手順は、ホストのブート後に一度だけ実行することができます。ただし、シングルユーザーモードでの実行を推奨します。

1. **DHCP** エージェントを設定して、サーバーと交換するパケットの詳細をログに記録することができます。この記録を行うには、以下のようにして、デバッグモードをオンに設定してエージェントを開始する必要があります。

```
/sbin/dhcpagent -n -d3 &
```

`-d3` フラグはレベル 3 でのデバッグをオンに設定し、`-n` フラグは「DHCP が正常な場合でもインタフェースを構成してはならない」という意味です。

注 - レベル 3 およびそれより下位のレベルでは、ユーザーに適切な情報が戻されます。レベル 3 より上位のレベルは、情報が生のパケットのまま戻されるため、開発者の方または高度な専門知識を持つ方だけが使用します。

2. dhcpagent のインスタンスは一度に **1** つだけが実行可能なため、ここでエージェントを開始する前に、すでに起動済みのエージェントをすべて停止する必要があります。エージェントを停止するには、エージェントのプロセス **ID** を調べて、以下のように終了シグナルを送信します。

```
kill -TERM process_id_of_dhcpagent
```

3. エージェントをデバッグモードで開始した後で、以下のように入力して、手動でインタフェースの構成を試行します。

```
ifconfig interface_name auto_dhcp
```

送受信されたパケットが表示されます。

注 - DHCP がインタフェースの構成を試行している間、インタフェースはパケットの送受信を行うことができません。インタフェースがダウンしている間は、その他のネットワークサービス (たとえば NIS や NFS) が悪影響を受ける場合があります。

▼ DHCP サーバーをデバッグモードで動作させるには

DHCP サーバーを停止して、デバッグモードで再起動します。以下に例を示します。

1. 停止スクリプトを使用してサーバーを停止します。

```
/etc/init.d/dhcp stop
```

2. サーバーをデバッグ・冗長モードで再起動します。ただし、/etc/init.d/dhcp 起動スクリプト内に指定されているフラグに加えて、-d フラグと -v フラグを使用します。たとえば、i オプションが存在する場合は、以下の形式でコマンドを入力します。

```
/usr/lib/inet/in.dhcpd -i interface_names -d -v
```

DHCP クライアントの再起動

DHCP クライアントをデバッグモードで動作させた後で、リブートを試行することができます。リブートを行うと、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。

▼ DHCP クライアントを再起動するには

- クライアントをリブートします。

▼ DHCP サーバーを再起動するには

1. **DHCP** サーバーにスーパーユーザーでログインします。
2. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

約 10 秒間待機します。

3. 以下のように入力します。

```
/etc/init.d/dhcp start
```

▼ デバッグの完了後に DHCP サーバーを再起動するには

1. **DHCP** サーバーのデーモンを再起動します。
2. **DHCP** サーバーにスーパーユーザーでログインします。
3. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

約 10 秒間待機します。

4. 以下のように入力します。

```
/etc/init.d/dhcp start
```

一般的な問題

この節では、DHCP に関して検出される可能性がある、より一般的な問題の一部と、それらの問題に対する処置について説明します。

問題

DHCP クライアントが DHCPDISCOVER または DHCPREQUEST というメッセージを送出しているが、DHCP サーバーが応答しない。

検証: サーバーマシンのコンソール出力を確認します。割り当てるべき IP アドレスがサーバーに残っていないことが考えられます。

解決策: より多くの IP アドレスを追加します。

検証: サーバーマシンのコンソール出力を確認します。クライアントが認識されていないことをサーバーが示している場合は、DHCP サーバーのデータベースがフラッシュして、その結果としてクライアントの認識に失敗していることが考えられます。

解決策: クライアント上の DHCP キャッシュファイルをすべて削除します。

1. **Ctrl** –C を入力してブートに割り込みます。

2. 以下のように入力して、キャッシュを削除します。

```
cd /etc/dhcp; rm interface_name.dhc
```

3. 以下のように入力して、初期化プロセスを再起動します。

```
ifconfig interface_name dhcp release
```

検証: サーバーマシンのコンソール出力を確認します。クライアントのネットワークに対するサポートが DHCP データベースに追加されていないことが考えられます。

解決策: dhcpconfig を使用して、クライアントのネットワークに対するサポートを追加します。

検証: クライアントが DHCP サーバーのネットワークとは別個のネットワーク上にあり、かつ BOOTP 中継エージェントがインストールされていないか、または設定されていません。

解決策: BOOTP 中継エージェントをインストールして設定します。さらに、リモートネットワークの netmasks(4) データベースにエントリを追加することが必要な場合があります。

問題

クライアントのログに、アドレスがすでに使用中であるというメッセージが記録される。

検証: アドレスが他で使用中であるかどうかを検査します。同じメッセージがクライアントのログに継続して記録される場合は、サーバーがアドレスを検査していないか、またはアドレスを拒否するクライアントのメッセージを無視しているかのいずれかが考えられます。 n オプション付きで in.dhcpd コマンドを使用していないことを検査して確認します。

解決策: サーバーが不良なアドレスを配付したかどうかを確認します。サーバーが誤動作しているか、または別のユーザーが同じアドレスを不法使用しているかのいずれかです。

問題

以下のエラーメッセージが表示される。

`DHCP renewal on` *interface_name* `failed`

(*interface_name* に対する DHCP の更新が失敗した)

検証: DHCP クライアントが、指定したインタフェースについてのリースを更新することができませんでした。

解決策: DHCP サーバーが適正に動作していることを確認します。

問題

以下のメッセージが表示される。

`Address of interface` *name* `has changed`

(インタフェース *name* のアドレスが変更されている)

検証: インタフェースのアドレスまたは状態が、DHCP エージェントが予期するものと異なっています。アドレスが手動で変更されたことが考えられます。

解決策: 解決策はありません。エージェントは、インタフェースを構成する試行を停止します。

支援の要請先

上記の方法を適用しても問題を解決できない場合は、Solaris のご購入先にお問い合わせください。最善のサービスを確実に受けるために、以下の情報をあらかじめご確認の上、お問い合わせください。

- 正確な、受け取ったエラーメッセージ (メッセージを受け取った場合)
- クライアント上で動作しているオペレーティングシステムのバージョン
- マシンの種類
- ネットワークの形式 (イーサネット、トークンリング、FDDI、PPP のいずれか)

DHCP サーバーの障害追跡

この節では、DHCP サーバーに発生する可能性がある問題について説明します。

ファイルの使用時

ネームサービスとして `files` を使用している際に問題が発生した場合は、以下の指示に従います。

問題

`/var/dhcp` ディレクトリにアクセスできない。そのディレクトリが存在しないか、または UNIX のファイル読み取り権を持っていない。

検証: 以下のコマンドを使用します。

```
ls -d /var/dhcp
```

解決策: DHCP サーバーがまだ構成されていません。`dhcpconfig` を実行します。

NIS+ の使用時

ネームサービスとして NIS+ を使用している際に問題が発生した場合は、以下の指示に従います。

問題

NIS+ ドメイン内にルートオブジェクトが存在しない。

検証: 以下のコマンドを入力します。

```
niscat -o org_dir
```

解決策: Solaris NIS+ 設定用のマニュアルを参照してください。

問題

root のアカウントに、*org_dir*オブジェクトの下でテーブルを作成するアクセス権がない。

検証: 以下のコマンドを入力します。

```
niscat -o org_dir
```

解決策: nischmod コマンドを使用して *table.org_dir.domainname.* に対するアクセス権を変更します。

問題

root のアカウントに、*org_dir* の下でテーブルを作成するアクセス権がない。通常これは、root のアカウントの主体名が *org_dir* オブジェクトの所有グループのメンバではないか、または所有グループが存在しないかのいずれかを表します。

検証: 以下のコマンドを入力して、所有グループ名を検索します。

niscat -o *org_dir*

解決策:

1. nisgrpadm -l ***group*** と入力して、グループのメンバーを確認します。
2. 現システムの主体名がグループ内にない場合は、nisgrpadm -a ***group principalname*** と入力して主体名を追加します。通常グループは **admin** です。グループが **admin** ではない場合は、dhcpconfig スクリプトを編集してグループを変更し、使用中のグループ名と一致させます。
3. /usr/lib/nis/nisctl -fg と入力して、変更が即時に行われるようにキャッシュをフラッシュします。

問題

ドメイン名が設定されていない。

検証: 以下のコマンドを入力します。

```
domainname
```

このコマンドにより空文字列が表示された場合、当該ドメインにはドメイン名が設定されていません。

解決策: domainname コマンドを使用して正しいドメイン名を設定します。ドメイン名の値を /etc/default ドメイン内に置きます。

問題

NIS_COLD_START ファイルが存在しない。

検証: サーバーシステム上で以下のコマンドを入力します。

```
strings /var/nis/NIS_COLD_START
```

解決策: NIS+ クライアントを 1 つ作成します。『Solaris NIS+ QuickStart』を参照してください。

問題

NIS+ を選択したが、サイトで NIS+ が動作していない。

検証: サーバーにログインして、以下のコマンドを入力します。

```
ps -ef | grep nis
```

NIS+ が動作している場合は、/usr/sbin/rpc.nisd -YB という出力に類似した出力が表示されます。

解決策: 以下のようにして、NIS+ サーバーを作成します。

1. クライアント上で、**NIS+** の **root** のマスターサーバーをドメイン用に設定します。たとえば、以下のようにします。

   ```
   /usr/lib/nis/nisserve -r
   ```

2. ローカルの /etc ファイルから **NIS+** テーブルを生成します。たとえば、以下のようにします。

```
nispopulate -F /etc
```

3. サーバー上で、**NIS+** が動作していることを確認します。たとえば、以下のようにします。

```
/usr/lib/nis/nisstat
nisls org_dir
niscat hosts.org_dir
```

ネームサービスとして NIS+ を使用できない場合

以下のエラーメッセージのうちのいずれかまたは両方が表示されます。

```
!!! warning !!! trailing dot ignored - use dns domain name
syntax
```

(!!! 警告 !!! 後尾のドットが無視された - DNS のドメイン名構文を使用してください)

```
Error 20 from NIS+; unable to use NIS+ as name service.
```

(NIS+ でエラー 20。NIS+ をネームサービスとして使用できない)

上記のメッセージは、NIS+ ドメイン内に該当する名前が存在しないか、または NIS+ ドメインが存在しないかのいずれかを表します。以下の情報を使用し、NIS+ の構成内のエラーを見つけ出して解決します。

問題

サーバーシステムのドメイン名の末尾は、ピリオド 1 つで終わる。

検証: nisdefaults コマンドを入力して、ドメイン名の末尾にピリオドが 2 つあるかどうかを確認します。

解決策:

1. /etc/defaultdomain ファイルを編集して、ドメイン名から末尾のピリオド **(.)** を削除します。

2. システムをリブートし、dhcpconfig スクリプトを再度実行します。

問題

ホスト名にドメイン名が含まれている。たとえば、ホストが myhost ではなく myhost.Faxco.COM と設定されている。

検証: nisdefaults コマンドを入力して、ドメイン名を含んでいるホスト名を 2 度表示します。

解決策:

1. ホスト名が間違っている場合は、sys-unconfig コマンドを入力し、構成設定値を削除してシステムを停止します。
2. システムをリブートし、ホスト名とドメイン名の適正な設定値を指定します。

問題

root のアカウントに、NIS+ ドメイン内の *org_dir* オブジェクトに対する作成のアクセス権がない。

検証: 以下のコマンドを入力します。

```
niscat -o org_dir
```

解決策: nischmod コマンドを使用して、*table.org_dir.domainname* に対するアクセス権を変更します。

ファイルのネームサービスを利用する際の入出力エラー

以下のエラーメッセージが表示されます。

File system I/O error *number* accessing file datastore.

(ファイルのデータストアにアクセスする際に入出力エラー *number* が発生)

上記のエラーメッセージを受け取った場合は、以下に示すエラーメッセージのリストを調べます。以下に示すエラーメッセージは、/var/dhcp 内のファイルのオープン、読み取り、書き込みのいずれかをオペレーティングシステムが試行した際に、オペレーティングシステムが返すエラーメッセージです。

問題

エラー番号 2 (ENOENT)。

検証: ファイルまたはディレクトリが存在しません。

解決策: dhcpconfig コマンドを入力してファイルまたはディレクトリを作成します。

問題

エラー番号 13 (EACCES)。

検証: ファイルまたはディレクトリにアクセスした際に、UNIX のアクセス権エラーが発生しました。

解決策: su コマンドを使用して UNIX のアクセス権を変更します。

ユーザーに DES 資格がない場合

問題

以下のエラーメッセージが表示されます。

The user *user* does not have DES credentials in the NIS+ name service.

(ユーザー *user* には、NIS+ ネームサービスにおける DES 資格がない)

検証: 現システムの root のアカウントは、有効なデータ暗号化規格 (DES) 資格を NIS+ cred テーブル内に持っていません。

解決策: nisaddcred コマンドを使用して、 root のアカウントの資格を追加します。コマンド行に、UNIX ネット名と NIS+ 主体名を入力する必要があります。

ドメイン Faxco.COM 内のシステム mercury の DES 資格を追加する方法を以下の例に示します。

```
nisaddcred -p unix.mercury@Faxco.COM\
-P mercury.Faxco.COM. DES Faxco.COM
```

このコマンドでは、root のパスワード (暗号化された秘密鍵を作成するために必要です) を求めるプロンプトが表示されます。

データストア内にテーブルを作成するアクセス権がない場合

以下のエラーメッセージが表示されます。

`You do not have permission to create the` *tablename* `table in the` *servicename* `data store.`

(*servicename* データストア内に *tablename* テーブルを作成するアクセス権がない)

テーブルをデータストア内に作成する際に問題が発生した場合は、以下の情報を検査します。

問題

`root` のアカウントに、*org_dir* オブジェクトの下でテーブルを作成するアクセス権がない。

検証: 通常これは、`root` のアカウントの主体名が *org_dir* オブジェクトの所有グループのメンバーではないか、または所有グループが存在しないかのいずれかを表します。

解決策:

1. `niscat -o` ***org_dir*** と入力して、所有グループの名前を確認します。
2. `nisgrpadm -l` **admin** と入力して、グループのメンバーを確認します。
3. 現システムの主体名がグループ内にない場合は、`nisgrpadm -a group` ***principalname*** と入力して主体名を追加します。
4. `/usr/lib/nis/nisctl -f g` と入力して、変更が即時行われるようにキャッシュをフラッシュします。

ネームサーバーを判定できない場合

DHCP サーバーの構成の際にネームサーバーを見つけることができない場合の解決策を以下に示します。

問題

`dhcpconfig` スクリプトで、サーバー名と IP アドレスが一致しなかった。

検証: コマンド getent hosts *name* を入力して、サーバーの IP アドレスを検索します。

解決策: hosts データベース内にエントリを作成します。

問題

dhcpconfig スクリプトが、サーバーの間違ったネームサービスを使用している。

検証: /etc/nsswitch.conf ファイル内の hosts エントリを調べて、IP アドレスの検索に使用されているネームサービス (xfn、files、nis、nisplus、dns のいずれか) を確認します。

解決策: /etc/nsswitch.conf ファイル内の hosts 命令を適正なネームサービスに変更します。nscd を停止して再起動します。

問題

dhcpconfig スクリプトがネームサービスを検査しなかった。

検証: /etc/nsswitch.conf ファイル内の [NOTFOUND=RETURN] 命令に先行するネームサービスが優先しています。指定されたネームサービスがエントリを見つけられなかった場合は、この命令の後に表示されているネームサービスはすべて検査されません。

解決策: /etc/nsswitch.conf ファイルから [NOTFOUND=RETURN] 命令を削除し、再度 dhcpconfig スクリプトを実行します。nscd を停止して再起動します。

DHCP テーブルの設定を試行した際のエラー

以下のエラーメッセージのうちの 1 つが表示されます。

The user *username* does not have permission to update the dhcptab table in the *servicename* resource.

(ユーザー *username* には、*servicename* リソース内の dhcptab テーブルを変更するアクセス権がない)

Error 10 from the Table subsystem accessing dhcptab table, message: NIS+ error while executing nis_modify_entry for *[key=SUNW.PCNFS.5.1.1,flag=m],dhcptab.org_dir.island.ocean.*: Permission denied
Error trying to set up DHCP table, exiting.

(dhcptab テーブルにアクセスした際の Table サブシステムのエラー 10 の場合のメッセージ。*[key=SUNW.PCNFS.5.1.1,flag=m],dhcptab.org_dir.island.ocean.* に対して nis_modify_entry を実行中にNIS+ のエラー: DHCP テーブルの設定を試行した際にアクセス権拒否のエラーが発生して終了した)

Error 10 from the Table subsystem accessing dhcptab table, message: NIS+ error while executing nis_modify_entry for *[key=SUNW.PCNFS.5.1.1,flag=m],dhcptab.org_dir.island.ocean.*: Object with same name exists Error trying to set up DHCP table, exiting.

(dhcptab テーブルにアクセスした際の Table サブシステムのエラー 10 の場合のメッセージ。*[key=SUNW.PCNFS.5.1.1,flag=m],dhcptab.org_dir.island.ocean.* に対して nis_modify_entry を実行中に NIS+ のエラー: DHCP テーブルの設定を試行した際に同じ名前のオブジェクトが存在するというエラーが発生して終了した)

上記のエラーメッセージのうちの 1 つを受け取った場合は、以下の情報を検査すれば、DHCP サーバーの構成中に DHCP テーブルの設定を試行した際に発生した問題の解決策があります。

問題

NIS+ または UNIX のファイルシステムから DHCP テーブルにエントリを追加するアクセス権を持っていない。

検証: アクセス権を検査して、DHCP テーブルに対する必要なアクセス権を設定します。

解決策: 管理者が所定の管理グループのメンバーであり、NIS+ のマスターサーバーに書き込むアクセス権を持っていることを確認します。

dhcp_network テーブルへのアクセス権がない場合

以下のエラーメッセージが表示されます。

You do not have permission to create {update} the *tablename* table in the *servicename* data store.

(*servicename* データストア内で *tablename* テーブルを作成する {変更する} アクセス権がない)

上記のメッセージを受け取った場合は、以下の情報を検査します。以下に示すのは、DHCP サーバーの構成中に dhcp_network テーブルにアクセスした際に発生した問題に対する解決策です。

問題

NIS+ または UNIX のファイルシステムから dhcp_network テーブルにエントリを追加するアクセス権がない。

検証: アクセス権を検査して、 dhcp_network テーブルに対する必要なアクセス権を設定します。

解決策: 管理者が所定の管理グループのメンバーであり、NIS+ のマスターサーバーに書き込むアクセス権を持っていることを確認します。

DHCP クライアントの障害追跡

DHCP クライアントの障害追跡を行う場合は、クライアントの構成とクライアント～サーバー間の通信についての問題点を理解しておく必要があります。DHCP がサーバーと通信できなかったか、または受け取った構成応答が間違っていたかのいずれかのために、DHCP がクライアントを正しく構成することに失敗する場合があります。さらに、クライアントが自己の IP アドレスを更新することができない場合には、DHCP のリースの期間の終わり頃になって問題が発生することがあります。

クライアントがサーバーと通信できない場合

クライアントとサーバーが相互に通信を行うことができない場合は、以前の DHCP のトランザクションからキャッシュに書き込んだ構成をクライアントが持っているかいないかに応じて結果が異なります。クライアントがキャッシュに書き込まれた構成を持っていて、かつリースがまだ有効な場合は、キャッシュのデータを使用してインタフェースを構成します。

ただし、キャッシュに書き込まれている構成が有効であるという外部での確認をクライアントが受け取っていないため、IP アドレス、ルーターのアドレス、およびその他の情報が有効であるという保証はありません。構成を受け取った際のネットワークとは別のネットワークにインタフェースが接続された場合は 2 種類のエラーが発生する可能性があり、発生する場合にはいずれか 1 つが発生します。その他のネッ

トワークサービスを開始した際にいずれかのエラーが表示される場合があります。あるいはネットワーク上のその他のホストとの通信ができない場合があります。

逆に、期限が切れていないリースを持つキャッシュが存在しない場合、インタフェースは構成されません。

受け取った DHCP 構成が無効な場合

以下の 2 つの理由から、構成が無効なことがあります。

1. クライアントに提供された IP アドレスが他で使用中であることを ARP によってクライアントが判定する。

   この場合、クライアントはサーバーに DHCPDECLINE メッセージを送信します。サーバーが不良なアドレスを 3 つ以上提供すると、`dhcpagent` は失敗します。

2. クライアントは IP アドレスを取得したが、確認を試行するとサーバーが確認ではなく DHCPNAK メッセージを送信する。

クライアントが DHCPNAK メッセージを 3 回以上受け取ると、`dhcpagent` は失敗します。このことは、サーバーが誤動作していることを示します。

問題をクライアントまたはサーバーに切り離す場合

問題をクライアントマシンまたはサーバーマシンのいずれかの問題として切り離す場合は、以下の処置を実行します。

問題

DHCP サーバーマシンが動作していない。

検証: クライアントと同じサブネット上の別のマシンにログインし、`ping` コマンドを使用してサーバーへの接続を試行します。

解決策: サーバーマシン上で問題を診断します。

問題

DHCP サーバーが動作していない。

検証: サーバーにログインして、以下のコマンドを入力します。

```
ps -ef | grep dhcp
```

解決策:

1. **DHCP** サーバーを停止して再起動します。

2. 以下のコマンドを入力します。

```
/etc/init.d/dhcp stop
```

3. 約 **10** 秒間待機してから、以下のコマンドを入力します。

```
/etc/init.d/dhcp start
```

問題

DHCP の際、ブートプロセスがハングする。

検証: インタフェースが *primary* とマークされますが、有効な DHCP トランザクションは発生していません。

解決策: `control-C` を入力して DHCP に割り込みます。ブートが継続します。

注 - ブートは継続しますが、ホストが接続されているネットワークに対しては、ホストの間違った構成が行われる場合があります。

クライアントが **DHCP** サーバーに接続できない場合

問題

DHCP クライアントのソフトウェアを構成し、そのクライアントを再起動した後で、ネットワーク上のサーバーにクライアントから接続することができない。DHCP のネットワークコマンドがすべて失敗し、DHCP サーバーへの接続をクライアントが試行したが失敗した、というメッセージが表示される。

一般的なエラーメッセージは以下のとおりです。

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

(DHCP サーバーまたは BOOTP サーバーが応答しない)

A request to access nonexistent dhcp_network database: *databasename* in datastore: *datastore*.

(データストア *datastore* 内の、存在しない dhcp_network データベース *databasename* へのアクセス要求である)

No more IP addresses for *network_address* network.

(*network_address* には、もう IP アドレスがない)

検証: 問題を正確に突き止めるために、以下の処置を実行します。

1. クライアントをデバッグモードで動作させます。
2. 手動でインタフェースの構成を試行して、ハードウェアが機能していることを確認します。
3. **DHCP** サーバーをデバッグモードで実行します。
4. snoop コマンドを使用して、**DHCP** サーバーとクライアントとの間で送信されるメッセージを追跡します。
5. 問題がクライアントマシン側にあるのかサーバーマシン側にあるのかを調べます。
6. エラーメッセージを調べて、以下の情報から解決策を選択します。

クライアントをデバッグモードで動作させる

クライアントをデバッグモードで動作させます。稼働している製品のマニュアルを参照してください。

Solaris クライアントの場合

1. 以下のように入力して起動します。

```
/sbin/dhcpagent -d3
```

DOS クライアントの場合

PC-NFS DOS クライアント上で、以下を実行します。

1. AUTOEXEC.BAT ファイルを編集し、SNCLIENT を SNCLIENT /D に置換します。
2. クライアントを再起動します。

インタフェースを手動で構成する

dhcpagent をデバッグモードで開始した後で、以下のように入力して、手動でインタフェースの構成を試行することができます。

```
ifconfig interface_name auto_dhcp
```

送受信されたパケットが表示されます。

サーバーをデバッグモードで動作させる

1. クライアントと同じサブネット上の **DHCP** サーバーにスーパーユーザーでログインします。
2. デバッグモードで、**DHCP** サーバーを終了して再起動します。たとえば、以下のようにします。

```
/etc/init.d/dhcp stop
/usr/lib/inet/in.dhcpd -d -v
```

あるいは、i オプションが存在する場合には、コマンドを以下の形式で入力します。

```
/usr/lib/inet/in.dhcpd -i interface_names -d -v
```

snoop を使用してネットワークのトラフィックを監視する

1. クライアントと同じサブネット上の **DHCP** サーバーまたは **BOOTP** 中継エージェントにスーパーユーザーでログインします。

2. snoop コマンドを使用して、ネットワークのトラフィックを追跡します。たとえば、以下のように入力します。

```
snoop -o /tmp/output udp port 67 または udp port 68
```

あるいは

```
snoop -o /tmp/output udp port bootps または udp port bootpc
```

インタフェースごとの引数が存在する場合は、その引数を加えます。

3. クライアントをブートして、サーバー上でネットワークのメッセージを監視します。

4. 以下のように入力します。

```
snoop -i /tmp/output -x 0 -v
```

このコマンドにより、パケットのトレースを参照することができます。

エラーメッセージを調べる

デバッグモードで in.dhcpd コマンドを実行した結果の出力を調べ、調べたエラーメッセージまたは状態を使用して、以下の情報から解決策を見つけ出します。

問題

以下のエラーメッセージが表示される。

```
Datagram received on network device: le0
```

(ネットワークデバイス le0 上でデータグラムを受け取った)

```
ICMP ECHO reply to OFFER candidate: ip_address disabling
```

(OFFER 候補 ip_address に対する ICMP ECHO の応答が無効にされた)

検証: DHCP サーバーは、クライアントに対して IP アドレスを提供する前に、その IP アドレスに対して ping コマンドを使用して当該アドレスが使用中ではないことを確認します。クライアントが応答した場合、その IP アドレスは使用中です。

解決策: 設定した IP アドレスがまだ使用中ではないことを確認します。

問題

以下のエラーメッセージが表示される。

No more IP addresses for *network_address* network

(*network_address* ネットワークには、もう IP アドレスがない)

検証: クライアントの dhcp_network テーブル内に、使用可能な IP アドレスがありません。

解決策: dhcpconfig コマンドを使用して、さらに IP アドレスを割り当てます。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 IP アドレスが、クライアントが配置されているサブネットに対するものであることを確認します。

問題

dhcp_network データベース内に *id_name* という不良なクライアント ID がある。

検証: dhcp_network テーブル内のクライアント ID (MAC アドレス) が間違っています。

解決策: イーサネットを使用している場合、クライアント ID は 01 の後にイーサネットアドレスが続きます。アドレス内のすべての文字が大文字になっていることを確認します。00 は、アドレスが割り当てられていないことを表します。

問題

以下のエラーメッセージが表示される。

Request to access nonexistent dhcp_network database: *database_name* in datastore: *nisplus_datastore*.

(データストア *nisplus_datastore* 内の、存在しない dhcp_network データベース *database_name* へのアクセス要求である)

検証: DHCP サーバーの構成時に、dhcpconfig スクリプトがサブネットの dhcp_network テーブルを作成しませんでした。テストネットワークとして、LAN (たとえば、サーバー 1 つとクライアント 2 つ) を設定して切り離した場合に起こることがあります。

解決策: dhcpconfig コマンドを使用して、dhcp_network テーブルと新規 IP アドレスを初期化します。

問題

以下のエラーメッセージを受け取る。

`Client` *client_id* `is trying to verify unrecorded address` *ip_address*`, ignored.`

(クライアント *client_id* が、記録されていないアドレス *ip_address* の確認を試行して無視された)

検証: このメッセージを受け取る場合は、考えられる理由が 2 つあります。

1. dhcp_network テーブルが削除されている場合に、このメッセージを受け取ることがあります。Solaris DHCP サーバーのみを使用している場合は、通常これが理由です。
2. データストア用に NIS+ が使用されていないことが原因で情報が共有されていない場合に、このメッセージを受け取ることがあります。サーバーがデータを共有していることを確認します。

異機種システムが混在しているサーバーのグループがある場合、このメッセージを無視します。

解決策: 以下のように入力して、クライアント上の古いキャッシュファイルを削除します。

```
ifconfig interface_name dhcp release
```

問題

DHCP は開始されているが、必要なネットワークサービスの一部が開始しない。

検証: DHCP サーバーが、必要な構成を供給していません。

解決策: サーバーが、必要なパラメータを送信しない理由を調べます。必要なパラメータを送信するようにサーバーを構成します。

問題

DHCP は開始されているが、特定のネットワークサービス (たとえば、NIS や NIS+) がエラーを報告するか、またはハングする。ホストが、ネットワーク上のその他のホストと通信を行うことができない。

検証: `dhcpagent` コマンドが、DHCP と通信を行うことができない (DHCP が使用可能でないため、と考えられます) ために、キャッシュに書き込まれているデータを使用しました。

解決策: キャッシュを削除します。以下のように入力します。

```
ifconfig interface_name dhcp release
```

キャッシュを削除しても、正しい構成を取得するという問題は解決されないため、ホストを手動で構成する必要がある場合があります。以下のように入力してトリガーファイルを削除することにより、ブート時に DHCP を無効にする必要があります。

```
rm /etc/dhcp.interface_name
```

問題

クライアントはブートして適正に動作するが、以下のメッセージが表示される。

`DHCP renewal on` *interface_name* `failed`

(*interface_name* に対する DHCP の更新が失敗した)

検証: DHCP は動作していますが、`dhcpagent` によりサーバーに接続してリースの延長を行うことができません。

解決策: サーバーが応答しない理由を調べます。`dhcptab` 内に設定されているルーターの値が間違っているか、またはクライアントのネットワークに対して期限が切れていることが理由である可能性があります。

問題

失敗した DHCP の更新についてのメッセージを受け取り、その後で以下のメッセージがコンソールに表示される。

`DHCP lease expired on` *interface_name*`: interface is now down`

(*interface_name* に対する DHCP のリースの期限が切れて、インタフェースは現在ダウンしている)

ネットワークサービスがこの時点でハングすることが考えられる。

検証: リースの期限が切れました。クライアントは複数回の試行を行いましたが、リースを延長することができませんでした。

解決策: サーバーが応答しない理由を調べます。クライアントを再起動します。

BOOTP 互換モードにおいて、一部のクライアントが DHCP サーバーからブートしない場合

問題

DHCP デーモンが BOOTP 互換モードで動作している (–b オプション)。

検証: BOOTP はリース時間を使用しません。DHCP サーバーは、BOOTP フラグが設定されている空きアドレスを探して、BOOTP クライアントに割り当てます。

解決策: BOOTP アドレスを割り当てます。dhcpconfig を使用して、デーモンのオプションを変更します。

NIS + 構成の問題の診断

以下の情報を使用して、NIS+ ネームサービスの構成内にある、ブート時にクライアントがサーバーにアクセスできないというエラーを修正します。

問題

dhcptab テーブル内で、クライアントに対してネームサービスが構成されていない。

検証: サーバーにログインして、以下のコマンドを入力します。

```
dhtadm -P | grep ip_address
```

NISdmain、DNSdmain、NISservs などのエントリを検査します。エントリに対して入力されているアドレスが適正であることを確認します。たとえば、以下のようになります。

```
# dhtadm -P | grep 129.148.3.129.148.3.m:Subnet=255.255.255.0:Router=129.148.3.11:
Broadcast=129.148.3.255:NISdmain=''island.ocean'':NISservs=129.148.3.3:
```

注 - 上記の行は、実際には 2 行に分かれないで 1 行で表示されます。

解決策: dhtadm を使用して、間違っているアドレスがあればすべて変更します。

問題

NIS+ を使用しているが、サーバーが NIS+ 互換モードで動作していない。NIS+ テーブルには *Nobody* カテゴリ用の読み取り権がないため、NIS クライアントが NIS+ テーブルに格納されている情報を読み取ることができない。

検証: 以下のコマンドを実行します。

```
nisls -l org_dir
```

このコマンドにより、.r---rmcdrmcdr--- のアクセス権が表示されます。

rpc.nisd デーモンに対して Y オプションが設定されているかどうかを検査します。たとえば、以下のように入力します。

```
ps -deaf | grep nis
```

解決策:

1. **NIS+** サーバーにスーパーユーザーでログインします。
2. 以下のコマンドを入力します。

```
/usr/lib/nis/nisserver -r -Y -d domainname
```

問題

デフォルトのルーターが間違っているため、クライアントが別のネットワーク上のサーバーに接続することができない。

検証: dhcptab テーブル内のルーターのシンボルの定義が本当にルーターであることを確認します。

解決策: dhtadm を使用して、テーブル内のルーターのシンボルを訂正します。

問題

NIS+ を動作させているが、NIS クライアントに対する DNS 転送がオンに設定されていない。

検証: 以下のコマンドを使用します。

```
ps -ef | grep rpc.nisd
```

–B オプションは、DNS 転送がオンに設定されて NIS が動作していることを表します。たとえば、以下のようになります。

```
/usr/sbin/rpc.nisd -B
```

解決策: DNS 転送を有効にして、NIS 互換モードで NIS+ サーバーを起動します。たとえば、以下のように入力します。

```
/usr/sbin/rpc.nisd -YB
```

ネームサービス構成の問題の診断

以下の情報を使用して、NIS ネームサービスの構成内にある、ブート時にクライアントがサーバーにアクセスできないというエラーを修正します。

問題

デフォルトのルーターが間違っているため、クライアントが別のネットワーク上のサーバーに接続することができない。

検証: dhcptab テーブル内のルーターのシンボルの定義が本当にルーターであることを確認します。デフォルトのルーターに問題がある場合は、サーバーベースのツールを用いて訂正を行います。

解決策: dhtadm を使用して、テーブル内のルーターのシンボルを訂正します。

問題

dhcptab テーブル内で、クライアントに対してネームサービスが構成されていない。クライアントに対しては、ネームサービスは DNS、NIS または NIS+ である必要があり、さらに必要なパラメータをクライアントごとに指定する必要がある。881

検証: クライアントの構成に関連する、ネットワークに固有なマクロを検査します。

1. サーバーにログインし、以下のコマンドを入力します。

```
dhtadm -P
```

2. クライアントのネットワークに一致するエントリを探します。

解決策: dhtadm を使用し、以下に従って、ネームサービス用のクライアントのマクロを訂正します。

クライアントが、ネットワーク上の最初のクライアントである場合は、以下のようにします。

1. dhtadm を使用してエントリを訂正します。

2. 次に、サーバー上で以下のように入力します。

```
/etc/init.d/dhcp stop,
/etc/rc3.d/S34dhcp start
```

さらに、クライアントを再起動します。

注 - クライアントに対して、ネームサービスの選択をサーバーが指定することはありません。サーバーは関連する情報を提供するだけです。クライアントが自己のネームサービスを選択します。

マクロの変更がクライアントに伝達されない場合

dhtadm を用いて、クライアントの 1 つまたは複数のマクロを変更しましたが、マシン上で変更が反映されません。たとえば、クライアントのルーターを変更しましたが、クライアントは依然として古いルーターを使用しています。

以下の情報を使用して、変更済みのクライアントのマクロが DHCP サーバー上に反映されないという問題を解決します。

問題

dhcptab テーブルに加えられた変更を読み込むための、DHCP サーバーの再初期化が行われていなかった。マクロ定義を変更するたびに、DHCP サーバーの再初期化を行う必要がある。

解決策: dhcpconfig に *rescan* オプションセットを使用します。あるいは以下に従います。

以下のようにして、DHCP サーバーの再初期化を行います。

1. **DHCP** サーバーにスーパーユーザーでログインします。
2. 以下のように入力します。

```
/etc/init.d/dhcp stop
```

3. 以下のように入力して、**DHCP** デーモンを再起動します。

```
/etc/init.d/dhcp start
```

付録 **A**

PCNFSpro 用の付録

障害追跡

以下の障害追跡手法は、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の詳細情報を取得する場合は、『*SolarNet PC-Admin Admininstrator's Guide*』を参照してください。

- 338ページの「デバッグモードでの実行」
- 338ページの「クライアントが DHCP/BOOTP サーバーとの接続に失敗する場合」
- 340ページの「SNC スクリプト」
- 344ページの「ログインおよびログアウト」

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、またはエラーメッセージが表示される場合は、まず再起動してみてください。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の名前に置換します。たとえば、`c:\pcnfspro\dhcp\pk0.bin` とします。

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動作させるには

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。「保存」を選択して構成を保存します。以下に類似す

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 サーバーまたは中継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、以下のようになります。

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コンベンショナルメモリを使い切った場合

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、「コンベンショナルメモリが足りない」、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 TSRs 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` コマンドを用い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。その後で Ping アプリケーションを使用します。

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 アプリケーションを使用します。

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 アプリケーションを使用します。

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、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、スクリプトが実行された後でマウント解除されます。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すること、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、プリンタクライアント、NetBIOS、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の詳細は、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 *user1user2 user3 user4 user5user6user7*

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 は、ログインスクリプトに対応するように命名する必要があります。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。その後で以下のようにします。

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て確認します。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、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、または 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マウント解除し、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 (イーサネット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の名前および IP アドレスがあります。

索引

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

あ

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

の

は

ひ

む

め

も

ゆ

よ

ら

り

る

れ

ろ

わ