OPEN LIB

OPEN LIB PROV
Engineering Guide

OPEN LIB PROV
エンジニアリングガイド

2014 年 5 月

オーエスエスブロードネット株式会社

第 0.6 版 2014 年 5 月

Printed in Japan

改版履歴

版数	改版年月日	変更内容	備考
0.1	2013/11/26	Edition5 Interrim 版。	加筆用
0.2	2014/01/20	第 1 章、第 2 章、付録 A、付録 B を執筆。	改良設計・検証のインプット。
0.3	2014/02/21	第 3 章、第 4 章を執筆。	改良のアウトプット。
0.4	2014/03/14	付録 C を執筆。	改良のアウトプット。
0.5	2014/04/20	OPEN LIB シリーズ全体の仕様整合を目的とした文言・内容の加筆修正。	改良のレビュー結果を反映。
0.6	2014/05/07	第 5 章 外部連携の加筆修正。	他エンジニアリングガイドの改版を反映。

参考文献

RFC2131 - Dynamic Host Configuration Protocol (DHCP)
RFC4388 – DHCP Lease Query
RFC 1350, 2090, 2347, 2348, 2349 - TFTP
http://www.isc.org/
RFC5256 - The Transport Layer Security (TLS) Protocol Version 1.2
Broadband Forum TR-069 Amendment 3

本書の目的

本書は、OPEN LIB PROV の導入設計・運用設定・保守に必要な情報をまとめたものです。
本書は、OPEN LIB に主体的に係わるパートナー各社とエンドユーザーの技術者および、OPEN LIB に興味を持って下さった全ての方々に対する情報開示を目的に作成されています。本書のインターネット上での再配布および部分的な流用は、個人・法人を問わず自由に行えますが、弊社に著作権の帰属する情報の二次利用に際しては、弊社の著作権を明示して下さい。

本書の対象読者

本書は、OPEN LIB PROV の導入設計・運用設定に従事する SE、保守を行う CE および、システム拡張を行うプログラマを対象にしています。
以下の技術に関する知識があると、本書の理解が一層容易になります。
DHCP, DNS, FTP/TFTP,/SFTP/FTPS HTTP/SOAP, スマートグリッド、IEC、DLMS/COSEM、ANSI、SSL、TLS、Web サービス、UDP、TCP/IP、Linux、Java、Firebird、MySQL

その他

OPEN LIB に関する技術的なご質問は、E-Mail により以下まで送信して下さい。
info@ossbn.co.jp

弊社の知的財産権に含まれない規格・技術の記述や情報の更新・バグに関し、弊社では一切の責任を負いませんのでご了承下さい。

目次

第1章　システム概要

1.1.　PROV の概念

OPEN LIB（OPEN LIBrary product series）は、業界標準の技術規格・仕様に基づく、様々な業務システムへの応用が可能なソフトウェア開発基盤製品シリーズです。OPEN LIB PROV（PROVisioning：以降「PROV」）は、OPEN LIB EA（Embedded Agent：以降「EA」）を組み込んだ端末機器を、各端末の設定ポリシーに基づき遠隔設定すると同時に、オンライン化した端末のアドレス等属性情報をセンター側の管理システムに登録する、プロビジョニングソフトウェアです。

PROV はヘッドエンド側に配置され、EA との通信により、組み込み対象の端末機器をプロビジョニングします。PROV によるプロビジョニングの対象は、ホームゲートウェイ・STB・スマート TV・USB ドングル等の情報端末、スマートメーター・コンセントレーター等の電力計測用機器、および、火災センサー・防犯センサー等の各種センサー機器です。

PROV のシステム構成イメージを図 1.11.1 に示します。

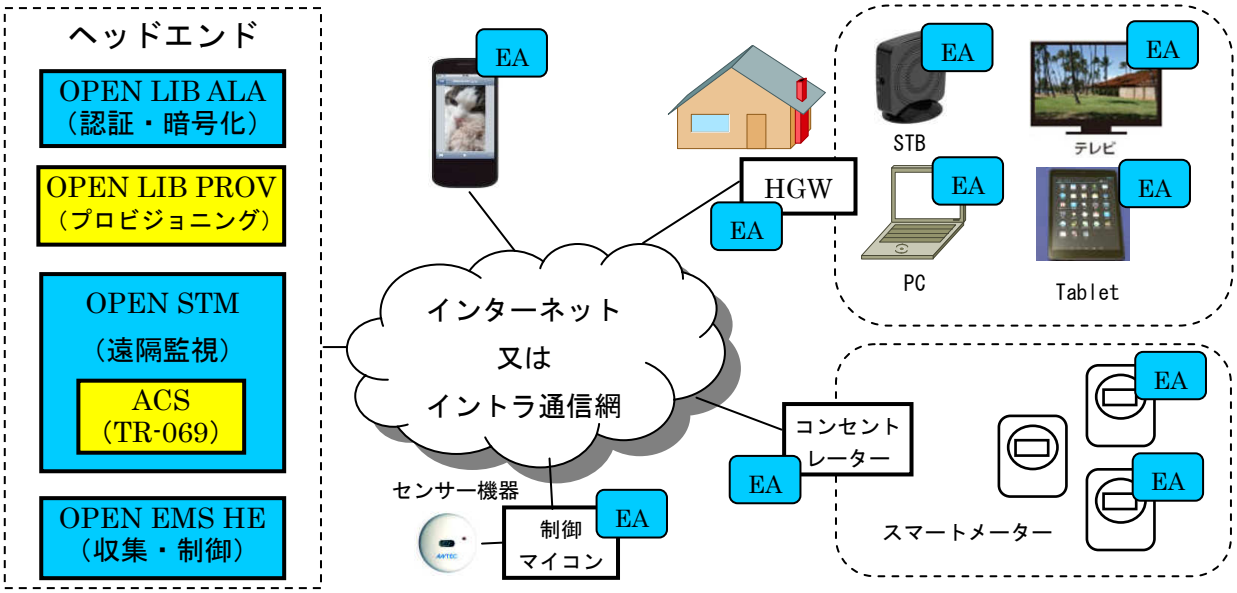


図 1.1.1　PROV のシステム構成イメージ

PROV のリファレンスモデルでは、端末機器の自動検知方式として、CATV 網の DOCSIS 端末プロビジョニング方式として広く使われている DHCPv4 の DHCP Discover による端末機器から上位向けのブロードキャスト通知を採用しています。

端末機器からのブロードキャスト通知を受信した PROV は、端末機器の IP アドレスと設定ファイル取得用の TFTP の IP アドレスを、DHCP Offer の返信により端末機器に通知します。

文書番号：OSSBN-EGMN-13-11-004

端末機器は、TFTP から取得した設定ファイルの記述内容に従い PROV とやり取りし、ファームウェア更新・時刻同期・レンジング等を順次進めます。各処理の正常終了と端末機器のオンライン化後、端末機器のアドレス＆属性情報が管理システムに登録され、プロビジョニングが完了します。

PROV のリファレンスモデルによるプロビジョニングのシーケンス例として、スマートメーターへの PROV 応用例を図 1.11.2 に示します。

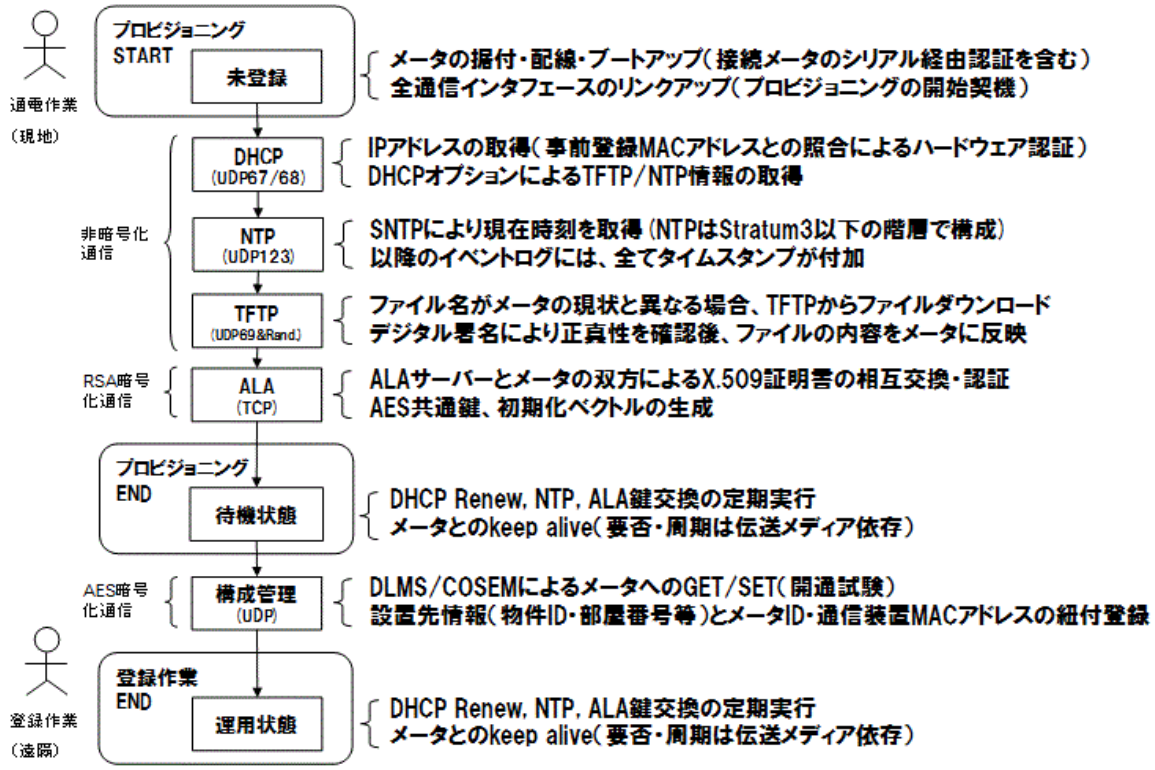


図 1.11.2 スマートメーターへの PROV 応用例

図 1.1.2 のうち、「非暗号化通信」として括られている DHCP, NTP, TFTP が、PROV リファレンス実装により提供されるサーバー機能群です。

PROV リファレンスモデルでは、IP アドレスや時刻、通信レベルで使用する情報のやり取りは暗号化の必然性に乏しい事、および、設定ファイルの転送はファイル自体へのダイジェスト付加＆暗号化が可能な事から、DHCP/NTP/TFTP を敢えて暗号化対象から外し、通信オーバーヘッドの削減を優先しています。

PROV リファレンスモデルでは、電力量や視聴情報などのアプリケーションレベルで使用する情報を、通信レベルで暗号化が必要な情報として再定義し、ヘッドエンドと端末機器間の通信を、共通鍵方式により暗号化します。また、共通鍵方式の暗号化に使われる秘密鍵は、ALA と端末機器間で、公開鍵暗号化方式により認証・暗号化された形で生成・更新されます。

PROV のシステム負荷抑制には、「プリプロビジョニング」が有効です。

プロプロビジョニングは、対象の端末機器 MAC アドレスと属性情報を DHCP サーバーに予め登録する方式であり、本来は機器設置・設定の作業効率の向上とトラックロールの短縮を目的とした手法ですが、未登録 MAC アドレスからの DHCP リクエストを無視する設定を行う事により、未知のアドレスからの大量リクエスト攻撃によるシステム全体への過負荷の回避に有効です。

DHCP は、ネットワーク内でブロードキャストの通過が許可される数少ない Well-known プロトコルの一つであり、端末機器の自動検知方式として有用ですが、端末機器がインターネットのように IP アドレス管理ポリシーの異なる複数ネットワークの集合に収容される構成では、ブロードキャストの通過がルーターないしは各ネットワークのエッジで拒否されるので使えません。

このため例えば OTT サービス等、インターネットの任意の場所に端末を接続する構成の場合、工場出荷設定や端末への手動設定等を組み合わせた別のプロビジョニング方式、例えば、OTT-STB 端末への手動設定による ACS-URL 通知や、工場出荷時の X.509 証明書設定＆センターへの事前登録等、DHCP&TFTP 以外の方法が必要になります。

TR-069 による OTT-STB のプロビジョニングシステム構成例を図 1.11.3 に示します。

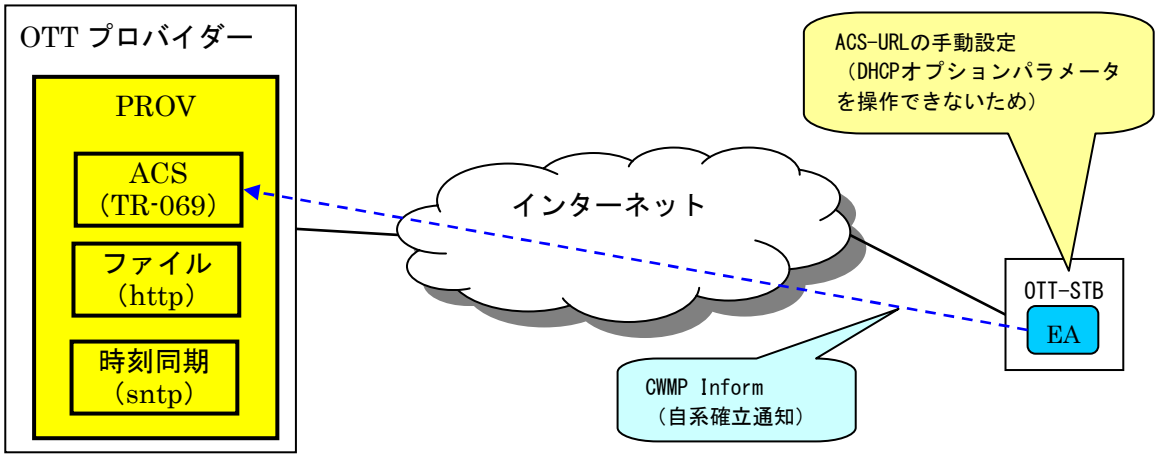


図 1.11.3 TR-069 による OTT-STB のプロビジョニングシステム構成例

TR-069 では、プロビジョニング通信に CWMP という HTTP/SOAP ベースのプロトコルが使用されます。CWMP のマネージャー機能が「ACS」です。TR-069 端末機器をプロビジョニングするには、ACS の IP アドレスまたは URL を OTT-STB に通知する必要があります。

TR-069 では ACS-URL の通知方式として、①DHCP オプションによる自動通知　②工場出荷時設定　③インストール時手動設定　の 3 つが定義されています。このうち①の DHCP 自動通知は、端末機器が接続される ISP の管理ポリシーに依存するため、特にインターネットを経由する場合、OTT プロバイダーによるオプション追加等の操作が困難です。このため上図では、③のインストール時手動設定を前提とした例を提示しています。

以上のように、TR-069 によるプロビジョニングシステムでは、DHCP が①②③のいずれかと ACS の組み合わせに置換され、更に、TFTP が HTTP に置換されます。

1.2. PROV のシステム構成

OTT サービスやスマートメータリングでは、端末機器の収容総数が数百万台を超える大規模なシステムが存在します。このような大規模システムで、各端末機器からのプロビジョニング要求タイミングが過度に集中すると、PROV が処理性能上のボトルネックとなり、端末動作の不安定やリブート時間の間延びが頻発し、システム全体の可用性低下を招く恐れがあります。一連のプロビジョニング動作で最もボトルネックになりやすい処理は、実行時間が比較的長く、タイミング重複が通信帯域の消費に直接的に影響するファイル転送です。

このため PROV のリファレンスモデルでは、ファイル転送プロトコルに処理負荷の最も軽い TFTP を採用しています。また、8 万 EA/PROV を収容設計上の目安とし、ネットワークエッジに PROV を分散配置する構成により、処理集中のボトルネックを物理的に分散させ、端末動作の安定性向上を実現しています。更に、分散配置される各 PROV を 1:1 構成で冗長化するシステム設計手法により、システムの可用性向上を図っています。

一般的に冗長構成は、ホットスタンバイ、ウォームスタンバイ、コールドスタンバイの 3 方式に分けられます。ホットスタンバイ方式は停止時間を極小化できますが、仕組みが複雑です。コールドスタンバイ方式は、障害検知時に O/S 起動・データ同期・アプリケーション起動を行うので、停止時間は長くなりますが、仕組みは単純です。

PROV の冗長構成は、両者の中間である 1:1 ウォームスタンバイ方式で動作します。

ウォームスタンバイ方式の予備 PROV は、待機中は O/S が起動状態で、HA(High Availability:高可用性)機能によりリース情報等の履歴データが常時同期されます。アプリケーションに相当する DHCP プロセスは、待機中は未起動状態で、フェイルオーバー動作時に起動されます。

DHCP などのプログラムがアプリケーションレベルで実装するフェイルオーバー方式は、複雑な制御により、動作上の不安定が生じる可能性があります。また、原則的に特定のアプリケーションプログラムに限定したフェイルオーバーであり、他のアプリケーションプログラムには無効です。

PROV のフェイルオーバー動作は、カーネルレベルでリース履歴を同期するため、DHCP アプリケーションは下位層のカーネルが行うフェイルオーバー動作を意識する事がなく、安定的に動作します。また、ACS 追加時にも同様に冗長化されるという利点があります。

（1）PROV リファレンスモデルのシステム構成

PROV リファレンスモデルのシステム構成を図 1.2 に示します。

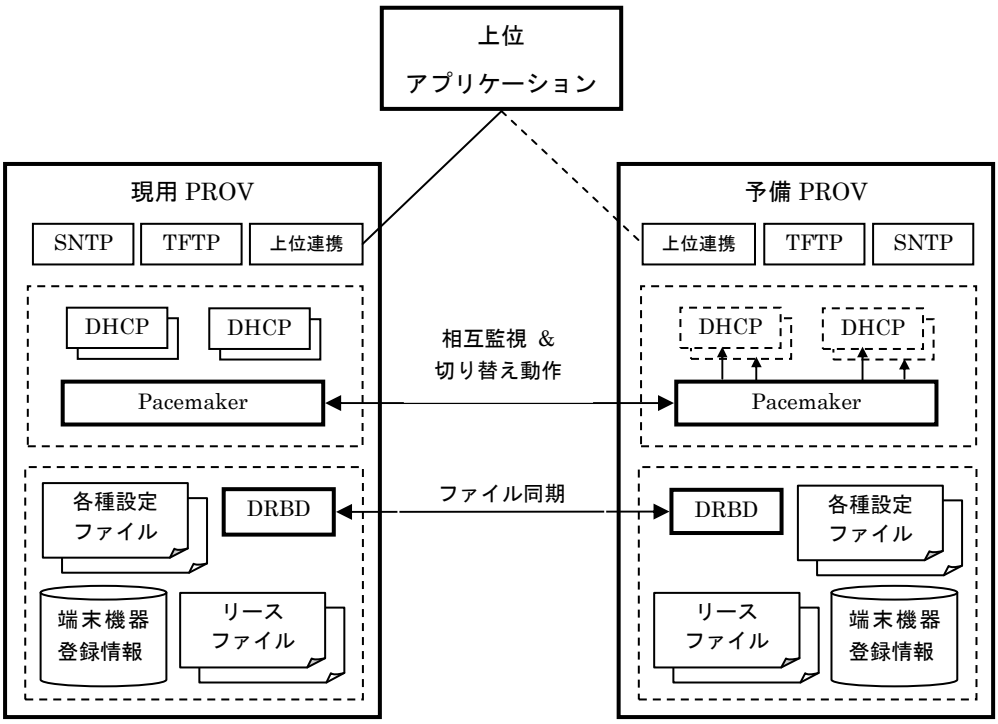


図 1.2 PROV リファレンスモデルのシステム構成

図 1.2 では、ウォームスタンバイ方式で 2 台の Linux PC サーバーを冗長化しています。

各種設定ファイル・端末機器登録情報・リースファイルは、DRBD により現用/予備 PROV 間で常時同期されます。DRBD のファイル同期により、予備 PROV への切り替え動作後も、同一端末機器による同一 IP アドレスの継続使用が保証されます。

現用 PROV に障害発生時、動作中の各 DHCP が、Pacemaker により予備 PROV に自動的に切り替わります。予備系 PROV から現用系への復旧操作は手動です。

TFTP は、現用/予備 PROV の双方で、DHCP と同一のネットワークインタフェース上で並列に動作させます。現用・予備 PROV の DHCP には、自身の TFTP をセカンダリ、相手側の TFTP をプライマリとして設定し、DHCP と TFTP とで負荷の分散を図ります。

なお、前節に示した TR-069 による OTT-STB のプロビジョニングシステム構成時、図 1.2 の DHCP が ACS に、TFTP が HTTP に置換されます。

（２）フェイルオーバー・フェイルバック動作

フェイルオーバー動作は、システムにより自動的に実行されます。フェイルオーバー動作の順序と所要時間は、以下の通りです。

①	現用 PROV の障害検知	18 秒
②	現用 PROV の各 DHCP プロセス停止	2 秒
③	予備 PROV の各 DHCP プロセス起動	10 秒

計 30 秒

リース履歴は、現用/予備 PROV の双方のファイルシステム間で同期されます。同期は常に実行されているため、所要時間には加算されません。

フェイルバック動作は､障害の復旧後､バッチスクリプトの一括実行により手動で行われます。フェイルバック動作の一括実行の順序と所要時間は、以下の通りです。

①	予備 PROV の各 DHCP プロセス停止	2 秒
②	現用 PROV の各 DHCP プロセス起動	10 秒

計 12 秒

（３）フェイルオーバーの動作条件

フェイルオーバーの動作条件は、以下の通りです。

- Heartbeat によるインターコネクト通信の途絶
  電源障害等によるシステム全体の動作停止、プロセスハングアップ、現用・予備間のケーブル断線、NIC 障害　など

- Linux watchdog デバイスへの書き込み失敗
  ディスク障害、メモリ障害、I/O コントローラ障害、ファイルシステム異常、カーネル動作不安定　など

- DOCS 端末用 DHCP ポートから各ネットワーク機器への ping 無応答
  他のネットワーク機器間のケーブル断線、NIC 障害　など

1.3. 機能一覧

PROV の機能一覧を表 1.3 に示します。

機能	分類	概要
DHCP	標準	ISC-DHCP
TFTP		Linux ディストリビューション標準　または　ATFTP
SNTP/NTP		Linux ディストリビューション標準
Syslog		同上
ディスク同期		DRBD
障害検知・自動切替		Pacemaker
ACS	オプション	OpenACS
HTTP ファイルサーバー		Linux ディストリビューション標準　または　Samba

表 1.3 機能一覧

PROV ではデフォルトの DHCP サーバーとして、ISC-DHCP を採用しています。

ISC-DHCP は、Internet Systems Consotium 社が開発するオープンソースソフトウェア DHCP であり、多くの Linux ディストリビューションに同梱され、手軽に使えるフリーの DHCP サーバーとして、Linux サーバー市場に広く普及しています。

TFTP には RedHat 系ディストリビューションで標準的な ATFTP を使い、マルチスレッド設定での運用を推奨しています。

HA（High Availability: 高可用性構成）には、DRBD&Pacemaker を使用します。

DRBD は、ネットワークを介して特定のパーティション＝ブロックデバイス（/dev/sda1 など）を自動的にミラーリングするソフトウェアで、2 台構成までのオープンソース版と、最大 4 台までの構成が可能な商用版があります。開発元は LINBIT 社(本社 :オーストリア)です。Linux のカーネル 2.6.33 以降では、オープンソース版 DRBD(8.3.7)が Linux カーネルの標準機能となりました。DRBD の関連情報は、http://www.drbd.org/から入手できます。

Pacemaker は、HA 化の対象となる各種リソースの監視・制御を行うソフトウェアであり、他のクラスター制御機能を持つソフトウェアと組み合わせて使用します。提供元は Linux-HA です。2008 年に、旧 Heartbeat の機能を部分的に取り込んだバージョン 1.0 がリリースされました。

Pacemaker の関連情報は、http://linux-ha.sourceforge.jp/wp/から入手できます。

1.4. 動作環境

動作環境を表 1.4 に示します。

項目	仕様
CPU	4 コア・1 ソケット構成以上、64bit CPU
RAM	総メモリ(GB) = 2 + クライアント数* 0.05 / 1,024
HDD	総ディスク(GB) = 30 + クライアント数 / 1,024
NIC	• インターコネクト通信と DRBD ファイル同期 * 2 (必須) • リモート保守, バックアップ、NTP * 1 (必須) • DHCP (含 DDNS), TFTP, Time * 1 (必須)
OS	Linux カーネル 2.6.32 以降・64bit 版
必須ソフト	• ISC-DHCP 4.2.3 以降 • Pacemaker 1.0.10 以降 • DRBD 8.3.1 以降

表 1.4 動作環境

1.5. 性能諸元

性能諸元を表 1.5 に示します。

No.	指標	性能（条件）	備考
1	最大クライアント数/PROV	80,000 クライアント	
2	ログ保持期間	90 日間	
3	DHCP 処理速度(Discover)	最大 300 トランザクション/秒　※	
4	DHCP 処理速度(Renew)	最大 600 トランザクション/秒　※	

表 1.5 性能諸元

※　TFTP による設定ファイル転送時間、TFTP 同時セッション数超過時の処理待ち時間、DHCP タイムアウトやエラー発生による再送時間等、環境に依存する各種のオーバーヘッドを除外し、DHCP シーケンスのみを前提とした、理論上の所要時間です。

第2章　動作原理

2.1.　ISC-DCHP の構造

2.1.1.　ISC-DHCP のシステム構造

ISC-DHCP は、DHCP サーバー(dhcpd)、CLI(omshell)、設定ファイル(dhcpd.conf)、リースファイル(dhcpd.leases)、プロセスファイル(dhcpd.pid)により構成されます。

設定ファイルが起動時に一度だけ読み込まれる静的な情報であるのに対し、リースファイルには起動時の読み込み後、最新の IP 貸出状態が継続的に都度追記されます。

動作中の DHCP サーバーに新たな DHCP クライアントを登録したい場合、CLI(omshell)を介して DHCP サーバー内のキャッシュ領域とリースファイルの双方を追記更新するか、ないしは、テキストエディタで各ファイルに新たな DHCP クライアントを登録後、対象の DHCP サーバーを停止・再起動します。

CLI を介してリースファイルに追記更新された設定内容は、操作の度に行が追加されます。削除行は delete フラグが付与される形で残り、DHCP サーバーの再起動時にリースファイルから実際に削除されます。

ISC-DHCP のシステム構造を図 2.1.1 に示します。

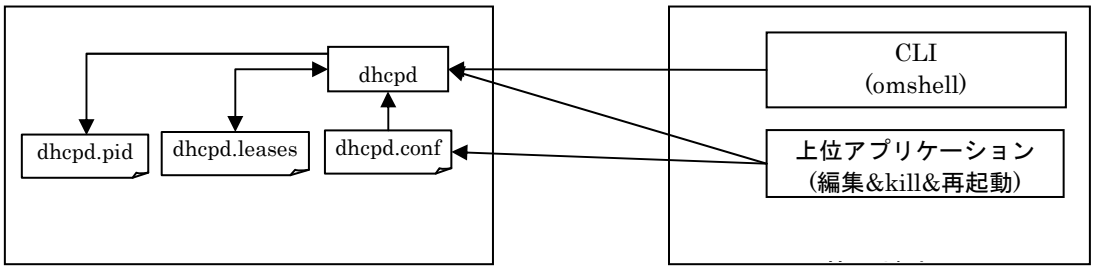


図 2.1.1 ISC-DHCP のシステム構造

CLI によりリモート環境から動的に設定を行う場合、TCP ポート 7911 がデフォルトで使用されます。CLI 用のポート番号は、設定ファイルに定義します。

DHCP サーバーは 67、DHCP クライアントは 68 を使用します。

プロセスファイルには、DHCP サーバーの pid が記録されます。DHCP サーバーを終了する場合、プロセスファイルを参照し、ファイル内に記録された pid を kill します。

フェイルオーバー構成時、DHCP サーバープロセスのキャッシュとリースファイルの情報は同期されますが、設定ファイルの情報は同期されません。

ISC-DHCP のフェイルオーバー動作仕様の詳細は、man を参照して下さい。

2.1.2. ISC-DHCP のデータモデル

ISC-DHCP のデータモデルを図 2.1.2 に示します。

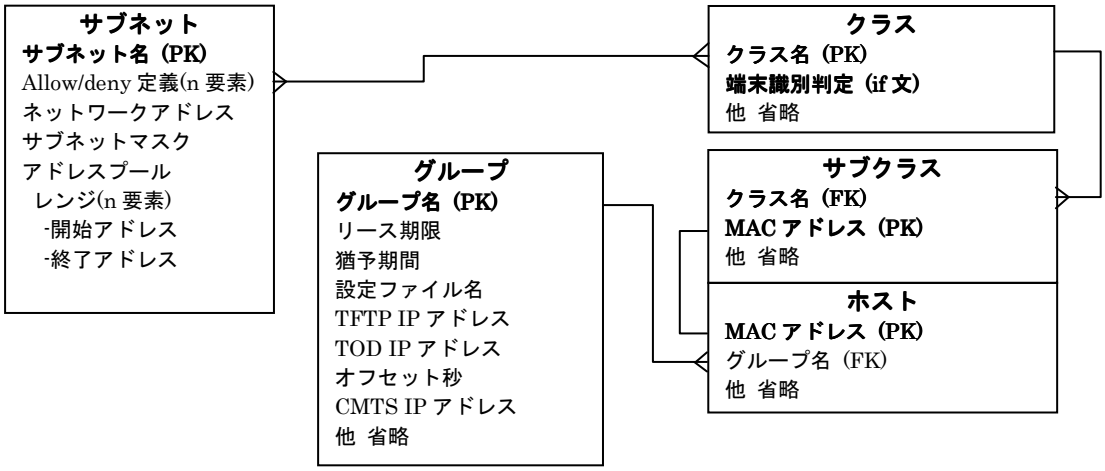


図 2.1.2 ISC-DHCP のデータモデル(ER 図)

サブネットは、、一つのネットワークアドレスと、複数のレンジから構成されます。レンジは、IP アドレスの範囲を定義します。サブネットの上位に、複数のサブネットを束ねる共有ネットワーク(shared-network)という概念もあります。

グループは、リース期限や各種オプションなど、DHCP クライアントの動作を定義します。

クラスは、DHCP クライアントのクラス分類概念です。端末種類を識別する if 判定文を含めると、DHCP オプション(82 など)から端末種類を自動的に識別できます。

サブネットの allow/deny 定義により、サブネットから特定のクラスを包含・除外できます。allow/deny 定義は、m:n 関係定義と論理的に類似なため、図中では m:n 関係の図記号で表現しています。

サブクラスとホストの組み合わせにより、DHCP クライアントを構成できます。運用では、対象クライアントのサブクラスとホストを一括して操作（追加・更新・削除）します。

ISC-DHCP は、IP 管理と動作設定管理のデータモデルが完全に分離されているため、クラスとグループ間、グループとサブネット間の定義に、循環参照のような依存関係はありません。また、設定ファイル中のグループ・クラス・サブクラス・ホスト宣言のいずれにも DHCP オプション等の細かい定義を追記できるなど、柔軟性に優れています。

設定内容に重複や矛盾がある場合の優先順位は、高い方から順に、

ホスト ＞ グループ ＞ サブネット ＞ 共有ネットワーク ＞ トップレベル および、

サブクラス ＞ クラス ＞ サブネット ＞ 共有ネットワーク ＞ トップレベル です。

矛盾が検出された場合、優先順位の高い方の設定内容で動作します。

2.1.3. ISC-DHCP の設定・運用方法

本節では説明を具体化するために、DOCS 端末プロビジョニングを前提にしています。

（１）設定ファイル(dhcpd.conf)の構造

設定ファイルには、DHCP サーバー(dhcpd)の設定情報が定義されます。

設定ファイルは ASCII テキスト形式であり、大文字小文字の区別はありません。

整形目的のタブや改行を自由に挿入でき、クォートの内部を除き、ファイルの任意の箇所にコメントを追記できます。コメントは”#”で始まり、行末で終わります。

設定ファイルは、パラメータ文と宣言文のリストから構成されます。

パラメータ(parameter)文は、何かをどの様に行うか（例: リース期間)、何かを行うかどうか(例: 不明クライアントへの IP 割当方針)、クライアントにどの様なパラメータを与えるか(例: ゲートウェイ IP=220.177.244.7)などを定義します。

宣言(declaration)文は、ネットワークトポロジー、クライアント登録、クライアントに割り当て可能な IP の範囲、複数パラメータのグループへの紐付けなどに使います。

ネットワークトポロジーの宣言文には、shared-network と subnet があります。サブネットの IP プールは、subnet の内部に range を記述して定義します。複数の range が必要な場合、pool{ }で囲みます。サービスを受けるサブネットや、DHCP サーバーが接続するサブネットには、動的割り当てを受ける IP の有無によらず、すべて subnet が必要です。DHCP サーバーは subnet により、対象 IP とサブネットの関係を認識します。

クライアントの端末種類によりネットワークやパラメータを設定したい場合、クラス(class)を使います。クライアントは、自分が所属するクラスの定義に従い動作します。端末種類は、クラス中の照合文(match statement)により定義します。

例えば、DHCP オプション 60 vender-class-identifier により CM と EMTA を識別するクラスは、以下の通りです。

```
class "cm" {
match if substring(option vendor-class-identifier,0,6) ="docsis";
#  オプションを列記（例: option tftp-server-name "10.105.101.2"; 等）
}
class "emta" {
match if substring(option vendor-class-identifier,0,8)="pktc1.0:";
vendor-option-space docsis-mta;
}
```

CM の DOCSIS バージョンによっては、CM 設定ファイルの内容が異なります。

CM を事前に登録する運用モデルでは、登録時に固有の CM 設定ファイル名を予め定義したクラス（ないしはグループ）を指定しますが、未登録 CM を最低 CoS/QoS 設定で一旦オンライン化し、後日の契約時か変更時に CM を正規に登録する運用モデルの場合、CM 設定ファイル名を動的に判断して紐付ける必要が生じます。

この場合、CM の DHCP オプション 60 に含まれる ASCII 文字列で、指定する CM 設定ファイル名を切り替える方法が有効です。例えば、"docsis1.0:xxxxxxx"は 1.0CM なので、1.0CM 用の CM 設定ファイルを指定します。

一部の機能・仕様が DOCSIS に準拠できず、二つの DOCSIS バージョンの中間にあるようなハイブリッド CM（プレ 3.0 など）では、特殊な内容の CM 設定ファイルが必要な場合があります。この場合、オプション 60 の xxxxxxx 部に 16 進エンコードで格納されるモデムケーパビリティで判断する事も可能です。モデムケーパビリティの詳細については、DOCSIS 規格文書である CM-SP-MULPIv3.0 の Annex C.1.3.1 と Annex D.1.1 を参照して下さい。

なお 1.0CM では DHCP オプション 60 が MAY 定義のため、CM が古い機種の場合、上記の判別方法が使えない場合があります。このような場合、CMTS リレーエージェントにより付加される DHCP オプション 82 を組み合わせ、端末種別を判定します。

判定には、DHCP オプション 82 のサブオプションである agent.remote-id を使用します。CMTS のリレーエージェントは、CM と CPE からの DHCP 要求を受信時、agent.remote-id に CM の MAC を"aa:bb:cc:dd:ee:ff"形式で格納し、DHCP サーバーに転送します。

DHCP サーバーは、agent.remote-id と要求元の MAC を照合し、同じ場合は CM 発、異なる場合は CPE 発の DHCP 要求と判断します。

DHCP オプション 82 から CM と CPE を判定する if 文の具体例を以下に示します。

```
match if option agent.remote-id = substring(hardware,1,6); # CM の場合
match if not option agent.remote-id = substring(hardware,1,6); # CPE の場合
```

これを応用し、substring(option vendor-class-identifier,0,6)に”docsis”の文字列が格納されず、かつ agent.remote-id と要求元 MAC が同じ場合、発信元を 1.0CM と判定できます。

ネットワークトポロジーとクラスを対応付けたい場合、allow/deny を使います。

```
subnet 10.0.0.0 netmask 255.255.255.0 {
pool {
allow members of "cm";
range 10.0.0.11 10.0.0.50;
}
pool {
allow members of "emta";
range 10.0.0.51 10.0.0.100;
```

```
}
```

端末 MAC を予め登録し、これに従いネットワークやパラメータを設定したい場合、定義済の各クラスにサブクラス(subclass)を追加します。サブクラスには、クライアントの MAC 指定が必須です。

```
subclass "cm"    00:15:96:27:a1:8c;
subclass "emta" 00:15:96:27:a1:8d;
```

特定端末のみに特殊な設定を行いたい場合、サブクラスに設定を追加します。

```
subclass "cm" 1:08:00:2b:a1:11:31 {
option root-path "samsara:/var/diskless/alphapc";
filename "/tftpboot/netbsd.alphapc-diskless";
}
```

既知のクライアントにのみ IP を動的に割り当てたい場合や、特定のクライアントに IP を静的に割り当てたい場合、クライアント毎に host を定義します。
前者の場合の定義例を以下に示します。

```
host 0011E3DD550D { hardware ethernet 00:11:E3:DD:55:0D; }
```

後者の場合の定義例を以下に示します。

```
host 0011E3DD550D {
hardware ethernet 00:11:E3:DD:55:0D; fixed-address 192.168.0.100;
#   任意の設定を列記（例: option host-name " host0011E3DD550D "; 等）
}
```

ホストをサブネットに allow/deny 定義すれば、ネットワークトポロジーとクライアントを直接対応付ける事も可能です。しかしながら、クライアントの個別属性がサブネットの宣言文に混入し、サブネットの共通性が損なわれるため、運用上は推奨できません。

ネットワークやクラスとは異なる区分で、まとまったクライアント群に対して共通の設定を割り当てたい場合、各 host 行に何度も定義すると、運用管理が煩雑です。
このような場合、グループ(group)が便利です。
”gold.conf”を設定ファイルとする”Gold”グループの定義例を以下に示します。

```
group Gold {
filename "/cmconfig/gold.conf";
option bootfile-name "/cmconfig/gold.conf";
}
```

次に、特定のホストに Gold グループを割り当てます。

```
host host000F666A7355 { hardware ethernet 00:0F:66:6A:73:55; group "Gold";}
```

未登録クライアントすなわち、host 定義がないクライアントへの振る舞いは、unknown-clients フラグにより定義します。例えば、特定サブネットから未登録クライアントを排除したい場合、deny unknown-clients を pool{}内の冒頭に記述します。

DDNS を使用する場合、以下の定義を追加します。

```
key "ossbn.co.jp" {
     algorithm hmac-md5;
     secret "1GPx/sFNPz4OU/NuspDqo……（省略）"; # DNS に登録した認証鍵
};
ddns-update-style interim;
zone ossbn.co.jp. {  #正引き指定
     primary 192.168.10.40; # プライマリ DNS の IP
}
zone 1.168.192.in-addr.arpa. {  #逆引き指定
     primary 192.168.10.40; # プライマリ DNS の IP
}
```

更に、トップレベルの以下の箇所を変更します。

```
ddns-update-style interim;
```

更に、上述のものと同じ認証鍵で、DNS 側にも DDNS 設定を行います。
DNS 側の DDNS 設定は、DNS のマニュアルを参照して下さい。

（２）設定ファイル例(dhcpd.conf)

```
### トップレベル定義 ###
authoritative; #説明省略
option wpad-curl code 252 = text;  #WPAD 設定
ddns-update-style none; # DDNS 使用時、none を interim に変更
option time-servers 192.168.0.12,192.168.0.11;
option time-offset -10800; # UTC との時間差(秒)
default-lease-time 21600;
max-lease-time 21600;
ping-check true ; #DHCP 通信シーケンスを参照
deny bootp ; #DOCSIS では bootp は使用しない
log-facility local7; #Syslog 出力の facility 指定
omapi-port  7911; # OMAPI のポート指定

### MTA 用オプション定義(構文のみ。定義の変更要) ###
option space docsis-mta;
option docsis-mta.dhcp-server-1 code 1 = ip-address;
option docsis-mta.dhcp-server-2 code 2 = ip-address;
option docsis-mta.prov-server code 3 = string;
option docsis-mta-encap code 122 = encapsulate docsis-mta;
option docsis-mta.kerberos code 6 = string;

### ローカルサブネット定義 ###
subnet 192.168.0.0 netmask 255.255.255.0 {
option routers 192.168.0.254;
option subnet-mask 255.255.255.0;
}

### グループ定義 ###
group basic_cm {
next-server 192.168.0.12,192.168.0.11; #TFTP サーバーIP
filename "basic_cm.cfg"; #CM 設定ファイル名
option dhcp.bootfile-name "basic_cm.cfg"; #CM 設定ファイル名
}
```

```
group test_mta{
next-server 192.168.0.12,102.168.0.11;
filename "mta_ss.bin";
option domain-name-servers 192.168.0.81,192.168.0.82;
option domain-name "example.com";
option host-name = concat (
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 1, 1))),2),
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 2, 1))),2),
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 3, 1))),2),
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 4, 1))),2),
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 5, 1))),2),
suffix (concat ("0", binary-to-ascii (16, 8, "", substring (hardware, 6, 1))),2) );
option docsis-mta.prov-server
00:06:77:74:74:73:73:31:04:76:6f:69:70:07:68:
6b:63:61:62:6c:65:03:63:6f:6d:02:68:6b:00;
option docsis-mta.kerberos 05:42:41:53:49:43:01:31:00;
}

### クラス定義 ###
class "docsis_cm" {
match if substring(option vendor-class-identifier,0,6) ="docsis";
spawn with option agent.remote-id;
option log-servers 192.168.0.12,192.168.0.11;
option domain-name "example.com";
option docsis-mta.dhcp-server-1 192.168.0.11;
option docsis-mta.dhcp-server-2 192.168.0.12;
}
class "docsis_mta" {
match if substring(option vendor-class-identifier,0,8)="pktc1.0:";
option log-servers 192.168.0.71,192.168.0.72;
vendor-option-space docsis-mta;
}
```

```
### ケーブル端末用共有ネットワーク定義 ###
shared-network cmts-ossbn1 Cable2/0 {
option domain-name                  "example.com";
option domain-name-servers          dns.example.com;

subnet 10.174.0.0 netmask 255.255.0.0 {
pool {
deny dynamic bootp clients;
deny unknown-clients;  # 未登録 MAC を除外
deny members of “docsis_mta";  #MTA を除外
allow members of “docsis_cm";  #登録済 CM が対象
option routers 10.174.0.1; # CMTS のゲートウェイ IP
option subnet-mask 255.255.255.0;
range 10.174.3.1   10.174.3.254;
range 10.174.4.1   10.174.4.254;
}
pool {
deny dynamic bootp clients;
deny unknown-clients;  # 未登録 MAC を除外
deny members of “docsis_cm";  #CM を除外
allow members of “docsis_mta";  #登録済 MTA が対象
option routers 10.174.0.1; # CMTS のゲートウェイ IP
option subnet-mask 255.255.255.0;
range 10.174.5.1   10.174.5.254;
}
}
}  #共有ネットワーク宣言文の終わり

### MAC 登録-1(ホストとグループの対応付け) ###
host 00e0.6f39.f988 {hardware ethernet 00:e0:6f:39:f9:88; group “basic_cm";}
host 00e0.6f58.4da4 {hardware ethernet 00:e0:6f:58:4d:a4; group “test_mta";}
host 00d0.5900.0001 {hardware ethernet 00:d0:59:00:00:01; group “basic_cm";}
host 0040.7b00.0002 {hardware ethernet 00:40:7b:00:00:02; group “basic_cm";}

### MAC 登録-2(サブクラスとクラスの対応付け) ###
subclass "docsis_cm” 00:d0:59:00:00:01;
subclass "docsis_cm" 00:40:7b:00:00:02;
subclass "docsis_cm" 00:e0:6f:39:f9:88;
```

```
    subclass "docsis_mta" 00:e0:6f:58:4d:a4;
```

（３）リースファイル(dhcpd.leases)

DHCP サーバーは起動時にリースファイルを必要とします。リースファイルが無いと DHCP サーバーが正常起動しない上、エラーも表示されません。

リースファイルはインストール時に自動作成されないので、存在しない場合には該当位置に、内容なしの空ファイルを作成する必要があります。

リースファイルには、DHCP サーバーによる貸出履歴が記録されます。

DHCP サーバーによるリースファイル出力例を以下に示します。

```
lease 192.168.1.6 {
    starts 6 2010/03/13 16:08:19; # リース開始日時(UTC)
    ends 6 2010/03/13 22:08:19; # リース終了日時(UTC)
    binding state active; # 現在の状態。active, free のいずれか
    next binding state free; # リース終了後の予定状態。
    hardware ethernet 08:00:27:XX:XX:XX; # クライアントの MAC
}
```

（４）プロセスファイル(dhcpd.pid)

プロセスファイルは、プロセス ID を管理するファイルです。DHCP サーバーの起動時、dhcpd の pid がプロセスファイルに書き込まれます。

プロセスファイルの詳細な説明は、man を参照して下さい。

2.1.4. DHCP サーバーの起動・終了

（１）起動

DHCP サーバーの起動方法と、指定可能なオプションを以下に示します。

```
dhcpd [-p <UDP port #>] [-d] [-f] [-4|-6]
      [-cf config-file] [-lf lease-file] [-pf pid-file]
      [-tf trace-output-file] [-play trace-input-file]
      [-t] [-T] [-s server] [if0 [...ifN]]
```

-p ポート番号　DHCP 標準のポート番号(UDP67)以外を使用時に指定
-d　　出力を標準出力(ディスプレイ)に行う(デバッグなどで使用)
-f　　フォアグラウンドで実行(非デーモンモード)
-4 | -6 DHCPv4 or v6 の指定
-cf　　設定ファイルを指定(無指定時はデフォルト位置)
-lf　　リースファイルを指定(無指定時はデフォルト位置)
-pf　　プロセスファイルを指定(無指定時はデフォルト位置)
-t　　設定ファイルのチェックのみ実行
-T　　リースファイルのチェックのみ実行

起動前に設定ファイルの内容をチェックし、誤りがある場合、適宜修正します。

```
/usr/sbin/dhcpd -t -cf /etc/dhcpd.conf
```

設定ファイルの確認後、DHCP サーバーを起動します。

```
/usr/sbin/dhcpd -cf /etc/dhcpd.conf -lf /var/lib/dhcp/dhcpd.leases -pf /var/dhcp/run/dhcpd.pid eth0
```

最後の eth0 は、DHCP サーバーを実行するインタフェースの指定です。指定しない場合、全インタフェースで DHCP サービスが提供されます。起動時のコンソールに exiting.と表示された場合、DHCP サーバーは実行されず、エラーの原因が exiting.の直前行に表示されます。

（２）終了と再起動

SIGHUP(kill -HUP)による再起動には対応していないので、終了後に再起動します。

終了は、cat プロセスファイル又は ps -ef | grep dhcpd により pid を確認後、対象の pid を kill します。

```
kill -TERM pid
```

2.1.5. DHCP リースクエリーへの対応

（１）RFC4388 リースクエリー

ISC-DHCP で RFC4388 リースクエリーを使用する場合、以下の情報を dhcpd.conf に書き込み、同機能の使用・不使用を選択します。

```
allow leasequery;
# deny leasequery;
```

なおリースクエリーは CPE 用 DHCP 向けの機能であり、DOCS 端末向けの DHCP には設定不要です。

RFC4388 リースクエリーのメッセージタイプや指定オプションの概要については、2.1.6(1)を参照して下さい。

（２）Cisco 独自仕様リースクエリーへの対応

ISC-DHCP を Cisco 独自仕様のリースクエリーに対応させるには、以下の手順でソースの該当箇所を修正後、プログラムをリビルドします。

```
//include/dhcp.h
****以下の行を探し、****
-----------------------------------------------------------------------------------------------
#define DHCPLEASEQUERY 10
#define DHCPLEASEUNASSIGNED 11
#define DHCPLEASEUNKNOWN 12
#define DHCPLEASEACTIVE 13
-----------------------------------------------------------------------------------------------
****以下の行に置換します****
-----------------------------------------------------------------------------------------------
#define DHCPLEASEUNASSIGNED 11
#define DHCPLEASEQUERY 13
#define DHCPLEASEKNOWN 14
#define DHCPLEASEUNKNOWN 15
#define DHCPLEASEACTIVE 16
#define DHCPUNIMPLEMENTED 17
-----------------------------------------------------------------------------------------------
```

//include/dhcpd.h
****以下の行を探し、****
---
void dhcpinform PROTO ((struct packet *, int));
void nak_lease PROTO ((struct packet *, struct iaddr *cip));

****以下の行に置換します****
---
void dhcpinform PROTO ((struct packet *, int));
void dhcpleasequery PROTO ((struct packet *, int));
void nak_lease PROTO ((struct packet *, struct iaddr *cip));

//server/dhcp.c
****以下の行を探し、****

"DHCPLEASEQUERY",
"DHCPLEASEUNASSIGNED",
"DHCPLEASEUNKNOWN",
"DHCPLEASEACTIVE"

****以下の行に置換します****

"DHCPLEASEUNASSIGNED",
"DHCPLEASEQUERY",
"DHCPLEASEKNOWN",
"DHCPLEASEUNKNOWN",
"DHCPLEASEACTIVE",
"DHCPUNIMPLEMENTED"

//server/dhcpleasequery.c
****以下の行を探し、****
---
if (lease == NULL) {
dhcpMsgType = DHCPLEASEUNKNOWN;
dhcp_msg_type_name = "DHCPLEASEUNKNOWN";
} else {
if (lease->binding_state == FTS_ACTIVE) {

```
dhcpMsgType = DHCPLEASEACTIVE;
dhcp_msg_type_name = "DHCPLEASEACTIVE";
} else {
dhcpMsgType = DHCPLEASEUNASSIGNED;
dhcp_msg_type_name = "DHCPLEASEUNASSIGNED";
}
}

/*
* Set options that only make sense if we have an active lease.
*/

if (dhcpMsgType == DHCPLEASEACTIVE)
{
```

```
****以下の行に置換します****
```

```
if (lease == NULL) {
dhcpMsgType = DHCPLEASEUNKNOWN;
dhcp_msg_type_name = "DHCPLEASEUNKNOWN";
} else {
if (lease->binding_state == FTS_ACTIVE) {
dhcpMsgType = DHCPACK;
dhcp_msg_type_name = "DHCPACK";
} else {
dhcpMsgType = DHCPNAK;
dhcp_msg_type_name = "DHCPNAK";
}
}

/*
* Set options that only make sense if we have an active lease.
*/

if (dhcpMsgType == DHCPACK)
{
```

2.1.6. OMAPI/OMSHELL による動的設定

V3 以降の ISC-DHCP には、動的に設定を更新できる OMAPI が組み込まれています。
OMAPI を使うと、DHCP サーバーを再起動せずに、設定を動的に DHCP サーバーに反映できます。
OMAPI による更新は、dhcpd.leases に書き出されます。

（１） 設定ファイルの記述
設定ファイルに、omshell からの接続を待ち受けるポート番号を指定します。
デフォルトポートは 7911 です。指定方法は以下の通りです。

```
omapi-port  7911;
```

（２） OMSHELL の起動
操作に先立ち、OMSHELL を起動します。起動方法は以下の通りです。

```
/usr/bin/omshell
server [DHCP サーバーのＩＰアドレス]
port 7911
connect
```

（３） ホストの起動
ホスト”abcdef”を追加します。操作方法は以下の通りです。

```
new host
set name = "abcdef"
set hardware-address = aa:bb:cc:dd:ee:ff
set hardware-type = 1
set ip-address = 192.168.1.100
create
close
```

操作の結果、以下の設定が dhcpd.leases に書き出されます。

```
host hname {
     dynamic;
     hardware ethernet aa:bb:cc:dd:ee:ff;
     fixed-address 192.168.1.100;
}
```

dynamic;とは、dhcod.conf で作成されたものではない、という意味です。

（４） ホストの更新

ホスト”abcdef”の設定を更新します。操作方法は以下の通りです。

```
new host
set name = "abcdef"
open
set ip-address = 192.168.1.200 ←更新したい内容
update
close
```

（５） ホストの削除

ホスト”abcdef”を削除します。操作方法は以下の通りです。

```
new host
set name = "abcdef"
open
remove
close
```

削除すると、以下の内容が dhcpd.leases ファイルに書き出され、宣言以前のホスト”abcdef”の定義が全て無視されます。

```
host hname {
  dynamic;
  deleted;
}
```

DHCP サーバーを再起動すると、無視されていた行が削除されます。

（６） 注意事項

ISC-DHCP のメーリングリストアーガイブには、OMAPI でサブクラスを操作できない症例が報告されていますが、2011 年 10 月現在、未解決です。

OMAPI の商用利用に際しては、①サブクラスの操作を禁止 ②事前に十分な動作検証を実施 の 2 点を徹底して下さい。

2.2. 冗長構成

2.2.1. DRBD

（１） アプリケーションの概要

DRBD (Distributed Replicated Block Device) は、Linux O/S 上で分散ストレージ構成による高可用 (HA) クラスターを構築時、複数ノード間でディスクを同期します。

DRBD は、カーネルモジュールとユーティリティーの 2 つのパッケージで構成されます。カーネルモジュールは、2.6.33 以降の Linux カーネルに含まれます。ユーティリティーは、カーネルモジュールの動作を制御し、DRBD を Pacemaker 等他のプログラムと連携する管理ツール及びシェルスクリプト群です。

冗長構成時、現用・予備の双方に DRBD のカーネルモジュールとユーティリティーをインストールし、双方に動作設定を定義します。

DRBD では、コピー元を「プライマリ」、コピー先を「セカンダリ」と呼びます。レプリケーションは、常にプライマリからセカンダリへの方向です。

プライマリノードの障害時、管理ツールがセカンダリノードをプライマリ状態にします。復旧時、再びプライマリに戻し、停止期間に更新されたブロックを再同期します。

セカンダリの障害時、プライマリのみに書き込む非接続モードに移行します。復旧時、接続モードに戻し、停止期間に更新されたブロックをセカンダリに再同期します。

（２） 設定内容

PROV 固有の設定内容を表 2.2.1 に示します。

設定項目	設定内容	備考
同期対象フォルダ	/var/db	
動作モード	単一プライマリモード	
レプリケーションモード	プロトコル C	
転送プロトコル	TCP(IPv4)	Pacemaker と共用
転送速度	100Mbyte/s	
スプリットブレイン通知	無効	
スプリットブレイン修復	手動	
ディスクフラッシュ	サポート	
ディスクエラー処理方針	I/O エラーを伝えない	

表 2.2.1 PROV 固有の設定内容

（３） 注意事項

システムの運用開始時、プライマリ→セカンダリの順に起動して下さい。

2.2.2. Heartbeat と Pacemaker

（１） アプリケーションの概要

Heartbeat は Linux-HA の中核機能です。Heartbeat v3 では、Heartbeat v2 の各リソース制御プロセス(CRM, Tengine, Pengine)が分離されて Pacemaker と名前を変え、これらを除いた残りのプロセス(CCM, RA)が、引き続き Heartbeat となりました。

Pacemaker は単体では動作せず、Heartbeat のサブプロセスとして動作します。

Heartbeat 単体でも、アプリケーションサービスの監視を除く簡単な HA クラスター環境を構築できますが、Pacemaker を組み合わせると、様々なリソースの状態監視や制御操作を行えます。Pacemaker との組み合わせ構成時、Heartbeat は各ノードの死活監視とノード間の通信を担当し、Pacemaker 同士の通信基盤として動作します。Heartbeat リンクは、プロセス消失・無応答を相互にチェックする事で、ノード全体の障害を検知します。一方で Pacemaker は、仮想 IP アドレスや Apache、MySQL 等、各アプリケーションサービスの動作状態を監視し、異常を検知時に対象ノードを制御します。

特定のアプリケーションに固有の監視・検知や制御操作の実装には、各アプリケーション用の RA(Resource Agent)と呼ばれるプログラムを Pacemaker に組み込みます。

RA は、Linux-HA の定める OCF 形式に従い記述されるシェルスクリプトです。

Linux-HA では、ping による特定 IP アドレスの死活監視、DRBD の動作監視や、Apache・MySQL 等、幾つかのアプリケーション向けに RA を定義・公開しています。

RA を作成する場合、以下の URL のガイドに従います。

http://www.linux-ha.org/doc/dev-guides/ra-dev-guide.html

作成した RA は、/usr/lib/ocf/resource.d/heartbeat に配置します。

Heartbeat の動作設定は、/etc/ha.d/ha.cf に定義します。

Pacemaker の動作設定は、crm コマンドを通じて定義します。設定が多い場合、予め設定内容をテキストファイルに書き出し、ファイル入力により一括設定します。コマンドから設定した内容は、/var/lib/heartbeat/crm/cib.xml に自動的に反映後、全ノードに同期されます。

PROV では､現用･予備の双方に Heartbeat v3 を含む Pacemaker パッケージをインストールし、動作設定を定義します。更に、DHCP プロセスの一斉 kill と起動を制御する RA を、crm から登録します。

（２） 設定内容

PROV 固有の設定内容を表 2.2.2 に示します。

項目	場所	内容	備考
ログファシリティ	ha.cf	local4	SYSLOG 出力設定
keepalive		3s	Heartbeat 間の keepalive 監視
deadtime		15s	keepalive 無応答からノード停止認知までの時間
auto_failback		オフ	自動フェイルバックを抑制
watchdog		/dev/watchdog	Heartbeat MCP から watchdog への書込停止時に O/S 再起動
ping/ipfail		deadping :15s	自 DHCP ポートの死活監視
DRBD 監視	cib.xml	Interval:3s, Timeout:15s	
stonith		無効	
DHCP プロセス制御	-	nrcmd server start | stop	RA により実装

表 2.2.2 PROV 固有の設定内容

（３） 注意事項

Linux 上で動作する ISC-DHCP では、殆どのディスクアクセスが Linux キャッシュに対して行われるため、ディスク障害が発生した場合にも、DHCP サービスは正常に動作し続けます。

しかしながら、このような不安定な状態が長く続くと、リースファイル破損によるデータ消失が発生し、フェイルオーバー・フェイルバック時にリース情報の整合性が失われる潜在リスクが生じます。

このようなリスクを回避するには、ディスク障害を検知してフェイルオーバーする RA の追加登録が有効ですが、フェイルオーバー動作に伴い、DHCP サービスの中断頻度も増えるので、DHCP クライアントに対するサービス可用性が相対的に低下するトレードオフがあります。

PROV では DHCP サービスの可用性向上を優先し、ディスク障害検知なしをデフォルト動作としています。

ディスク障害を検知する RA を登録する場合、以下の URL を参考にして下さい。

http://sourceforge.jp/projects/linux-ha/wiki/hb-diskd

Heartbeat は時刻に敏感なアプリケーションです。冗長構成時、ノード間で NTP を使い、時刻同期を行うようにして下さい。

2.2.3. フェイルオーバー動作

DHCP のフェイルオーバー動作を図 2.2.3 に示します。

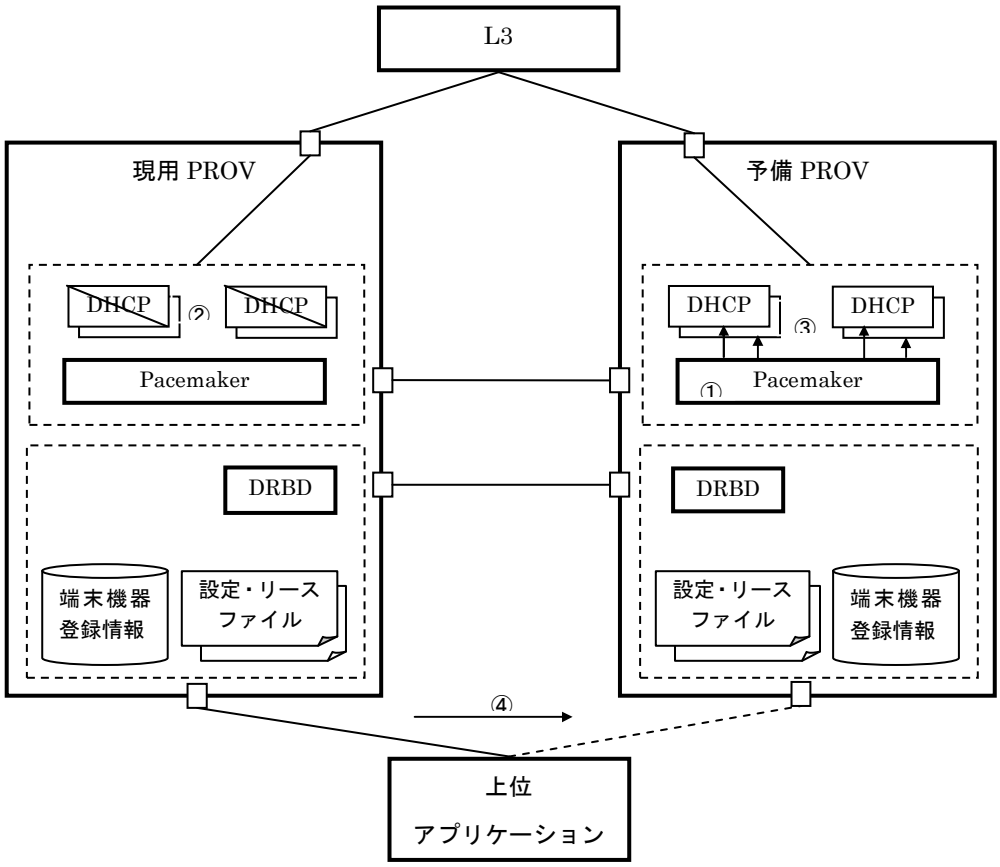


図 2.2.3 DHCP のフェイルオーバー動作

各動作の内容は以下の通りです。

① 障害発生中フラグを立てます。フラグが立っている間は Primary に戻りません。
② 現用 PROV の全 DHCP プロセスを kill します。
③ 予備 PROV の全 DHCP プロセスを起動します。
④ 上位アプリケーションの接続先を手動で切り替えます。

2.2.4. フェイルバック動作

フェイルバックは手動でのコマンド操作で実行します。
DHCP のフェイルバック動作を図 2.2.4 に示します。

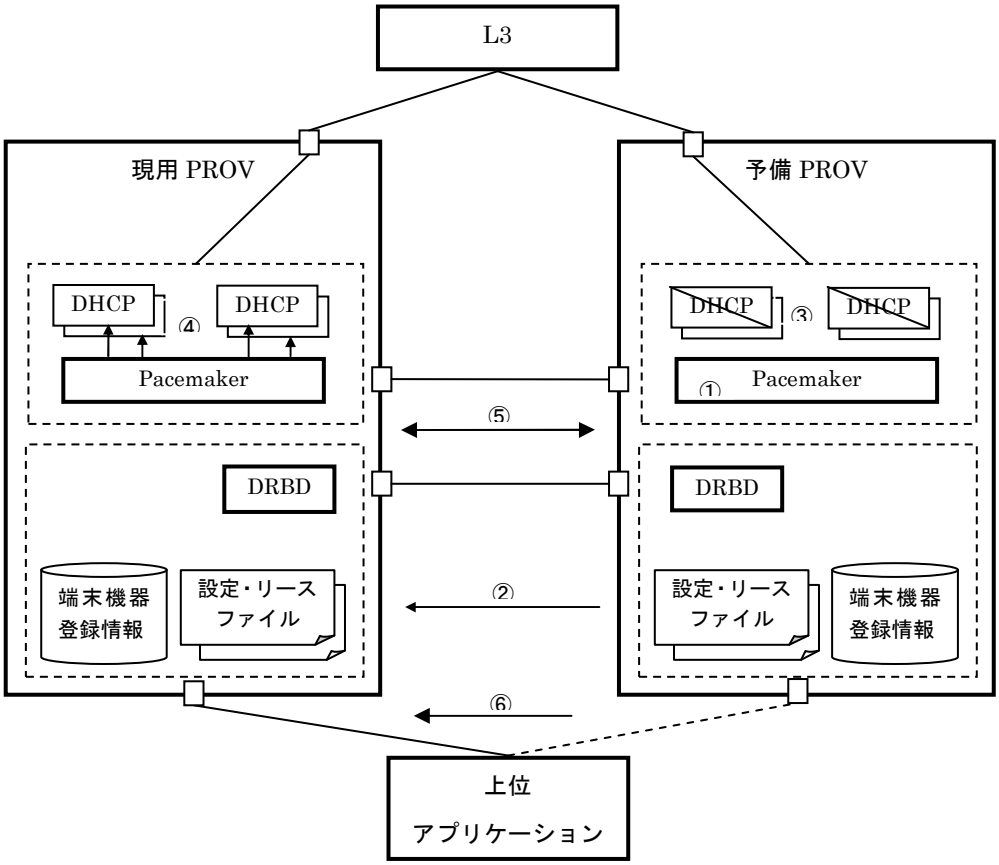


図 2.2.4 DHCP のフェイルバック動作

各動作の内容は以下の通りです。

① 障害発生中フラグを戻します。
② 設定・リースファイル、端末機器登録情報を復旧（再同期）します。
③ 予備 PROV の全 DHCP プロセスを kill します。
④ 現用 PROV の全 DHCP プロセスを起動します。
⑤ Pacemaker の監視を再開します。
⑥ 上位アプリケーションの接続先を手動で切り替えます。

①～⑤は、バッチ処理で一括実行されます。

第3章　PROV の導入

3.1.　設計・導入フロー

PROV の設計・導入フローを図 3.1 に示します。

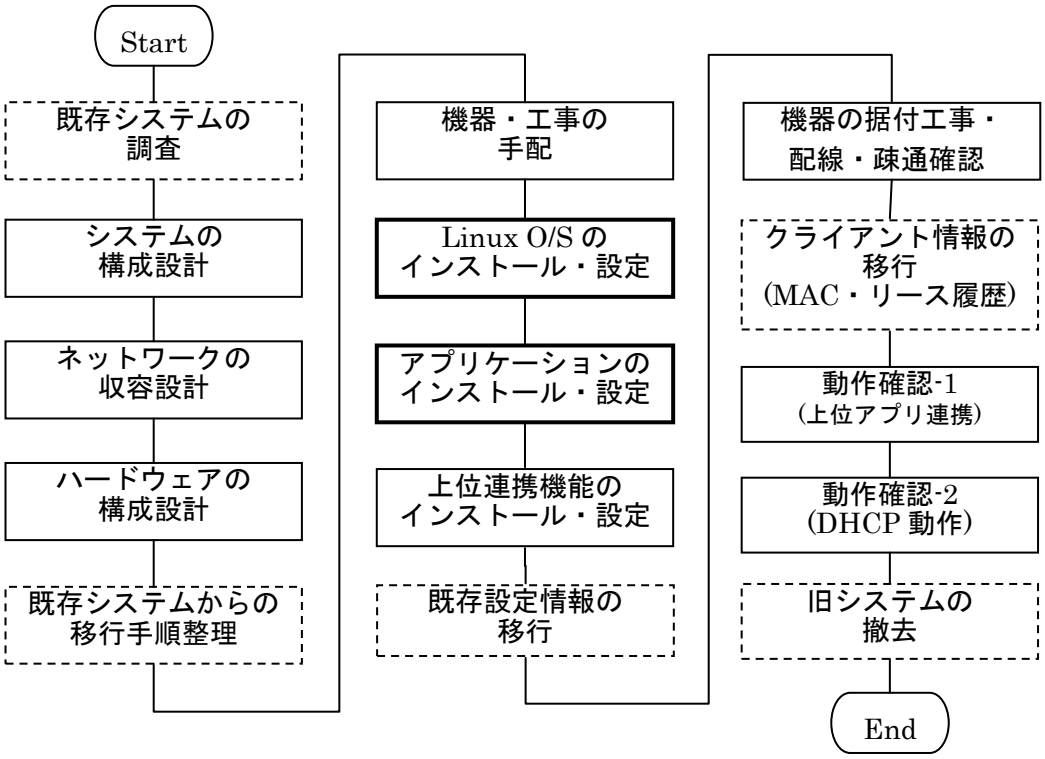


図 3.1 PROV の設計・導入フロー

破線の作業は、既存システムを PROV に置換する場合のみ発生します。

本章では、図中の太線枠の作業について説明します。

太線枠以外の作業については、第 1 章を参照して下さい。

機器の据付工事・配線・疎通確認については、機器メーカーの仕様書・マニュアルを参照して下さい。

3.2. Linux OS のインストール・設定

本項では、PROV 固有の内容のみ説明します。その他の一般的な内容については、使用する Linux ディストリビューション・ディスク環境の設定指針に従って下さい。

3.2.1. ディスク設定

PROV では、フォルダ/var/db が DRBD の同期対象となります。
ここでは LVM(Logical Volume Manager, 論理ボリュームマネージャー)の利用を前提に、HDD の空き領域に DRBD 用 LVM パーティションを作成します。

物理ボリュームを作成します。

```
#> pvcreate /dev/sda3
Physical volume "/dev/sdda3" successfully created
```

ボリュームグループを作成します。

```
#> vgcreate vg_r0 /dev/sda3
Volume group "vg_r0" successfully created
```

作成したボリュームグループから論理ボリュームを切り出します。

```
#> lvcreate -L 20G -n lv_r0 vg_r0
Logical volume "lv_r0" created
```

ファイルシステムの作成と、論理ボリュームへのフォルダ/var/db のマウントは、DRBD のインストール・設定後に行います。

動作設定のバックアップ用フォルダを作成します。

```
#> mkdir /etc/peer
```

次に、/etc/lvm/lvm.conf 内の filter オプションを以下のように設定します。

```
filter = ["a|sd.*|", "a|drbd.*|", "r|.*|"]
```

また、ライトキャッシュ無効にするため以下を設定します。

```
write_cache_state = 0
```

最後に、/etc/lvm/cache/.cache を削除して、古いキャッシュを削除します。

3.2.2. ネットワーク設定

PROV のネットワークインタフェース一覧を表 3.2.2 に示します。
備考欄の各 IP は、説明用の仮設定です。実際の設定は、設計方針に従って下さい。

名前	用途	ネットワーク	備考(特殊設定等)
eth0	上位アプリ接続 保守、バックアップ NTP (tcp/udp123)	管理 LAN	現 用 :192.168.10.1, 予 備 :192.168.10.2 (仮)
eth1	Pacemaker 通信	PROV 間直結 (クロスケーブル)	現用: 192.168.0.1, 予備: 192.168.0.2 (仮) 全て allow
eth2	Pacemaker 通信 DRBD 通信	PROV 間直結 (クロスケーブル)	現用: 192.168.1.1, 予備: 192.168.1.2 (仮) 全て allow
eth3	DHCP, DDNS TFTP (udp69) Time (tcp37)	サービス LAN	現 用 :192.168.20.11, 予 備 :192.168.20.12 (仮) 全て deny、DHCP, TFTP, Time, Log を allow

表 3.2.2 PROV のネットワークインタフェース一覧

DRBD と Heartbeat はホスト名を使用するため、ホストネームと IP アドレスの名前解決ができるように、/etc/hosts ファイルまたは DNS サーバーに両ノードのホストネームを登録します。

ネットワークパラメータをチューニングします。
/etc/sysctl.conf に以下の設定を追加し、UDP 受信バッファのサイズと、ルーティングキャッシュのガベージコレクションの頻度を拡張します。

```
net.core.rmem_default = 1290240
net.core.rmem_max = 1290240

net.ipv4.neigh.default.gc_thresh1 = 65536
net.ipv4.neigh.default.gc_thresh2 = 131072
net.ipv4.neigh.default.gc_thresh3 = 262144
```

3.2.3. Syslog 設定

ISC-DHCP・TFTP・Pacemaker・DRBD は、syslogd を利用してログを出力します。各アプリケーションのログ出力先を表 4.2.1(3)に示します。

アプリケーション	facility	priority	出力先
ISC-DHCP	Local6	info	/var/log/dhcp.log
TFTP	Local5	同上	/var/log/tftp.log
Pacemaker	Local3	err	/var/log/pacemaker.log
DRBD	Kernel	同上	/var/log/messages
ISC-DHCP	Local2	info	/var/log/v4cpe_dhcp.log
ISC-DHCP	Local1	info	/var/log/v6cpe_dhcp.log

表 4.2.1(3) 各アプリケーションのログ出力先

各出力ログの facility は、各アプリケーションの設定ファイルに定義します。

ログ出力ポリシーは、/etc/rsyslog.conf に定義します。

各アプリケーションからの/var/log/messages への出力を抑制し、各アプリケーションのログ出力先を定義します。太字の箇所が追加定義です。

*.info;mail.none;authpriv.none;cron.none**;local6.none;local5.none;local3.none;local2.none;local1.none** /var/log/messages

local6.*　　/var/log/dhcp.log
local5.*　　/var/log/tftp.log
local3.*　　/var/log/pacemaker.log
local2.*　　/var/log/v4cpe_dhcp.log
local1.*　　/var/log/v6cpe_dhcp.log

local5 は、将来的な DRBD 用のファシリティです。

ログローテーションポリシーは、/etc/logrotate.conf に定義します。

ログファイルを日単位に生成・置換し、90 日間保持する場合、logrotate.conf の内容を太字のように変更します。

daily
rotate **90**

3.2.4. sysstat のインストール

障害発生時の原因究明の為、sysstat ツールをインストールします。
まず、rpm コマンドでインストールされているかを確認します。

```
#> rpm -qa | grep sysstat
sysstat-9.0.4-18.el6.i686
```

インストールされていない場合、次のコマンドでインストールします。

```
#> yum install sysstat
```

インストール後、設定ファイル（/etc/sysconfig/sysstat）を編集し、ログの保存期間を 90 日に変更します。

HISTORY=90（初期値は 7）

3.2.5. watchdog のインストール

稼働しているノードが不安定になったときに、ホストを OS ごと再起動する機能である watchdog をインストールします。まず、rpm コマンドでインストールされているかを確認します。

```
#> rpm -qa |grep watchdog
watchdog-5.5-10.el6.x86_64
```

インストールされていない場合、次のコマンドでインストールします。

```
#> yum install watchdog
```

自動的に起動するように設定します。

```
#> chkconfig watchdog on
```

3.2.6. カーネル・パニック発生時の OS 再起動の設定

カーネル・パニック発生時に自動で再起動するよう/etc/sysctl.conf に kernel.panic パラメータを設定します。

kernel.panic には 0 以外の数値（発生時に自動再起動するまでの秒数）を設定します。

```
kernel.panic = 60  (/etc/sysctl.conf ファイルの末尾に追加)
```

ここでは 60 秒を設定しています。

設定変更後、サーバを再起動して、/proc/sys/kernel/panic ファイルの値が 60 になっていることを確認（cat /proc/sys/kernel/panic）して下さい。

3.2.7. コア・ファイルの出力設定

起動中のプロセスが異常終了した場合にそのコア情報を記録する為の設定を行います。
まず、/etc/profile.d/配下に ulmit.sh ファイルを作成します。

```
#> vi /etc/profile.d/ulimit.sh
```

続いて ulimit.sh に次の一行を追加します。

```
ulimit -S -c unlimited
```

ファイルを保存、終了後、/var/core ディレクトリを作成します。

```
#> mkdir /var/core
```

最後に/etc/sysctl.conf を開き、末尾に次の一行を追加します。

```
kernel.core_pattern = /var/core/%t-%e-%p-%c.core
```

3.2.8. SNMP エージェントのインストール・設定

SNMP 監視を行う場合、SNMP エージェントをインストールします。

次のコマンドで net-snmp がインストールされているかを確認します。

```
#> rpm -qa | grep net-snmp
```

インストールされていない場合、パッケージを指定してインストールします。

```
#> yum install net-snmp-x.x.x
```

続いて設定ファイルを編集します。

```
#> vi /usr/local/share/snmp/snmpd.conf
```

＜セキュティ設定＞

```
#       sec.name  source          community
com2sec local localhost private
com2sec mynetwork 192.168.0.0/24 public  #アクセス可能なネットワークを設定
```

＜アクセス・グループの設定＞

```
# groupName securityModel securityName
group MyROGroup v1 mynetwork
group MyROGroup v2c mynetwork
group MyROGroup usm mynetwork
```

＜アクセス範囲の設定＞

```
#       name           incl/excl     subtree         mask(optional)
view all    included  .1                               80 #この例では全て許可
```

＜アクセス許可グループの設定＞

```
#       group          context sec.model sec.level prefix read   write  notif
access  MyROGroup ""      any       noauth    exact  all none none
```

＜ディスク容量＞

```
disk / 10000  #環境に合わせて編集
```

編集・保存後、snmpd を再起動します。

```
#> /etc/init.d/snmpd restart
```

起動したことを確認後、外部サーバから MIB 情報にアクセスできる事を確認しておきます。

- snmpwalk -v 1 localhost -c public laTable # CPU 負荷情報
- snmpwalk -v 1 192.168.0.181 -c public diskTable # ディスク情報
- snmpwalk -v 1 192.168.0.181 -c public memTotalFree #メモリ情報

最後に自動起動設定を行います。

```
#> chkconfig snmpd on
```

3.3. アプリケーションのインストール・設定

3.3.1. ISC-DHCP

以下の URL からダウンロードします。
http://www.isc.org/

尚、ここでは安定動作版の dhcp-4.1-ESV-R2.tar.gz を使用します。バージョンは都度確認して下さい。

ダウンロードした tar アーガイブを解凍します。

```
#> tar xzvf dhcp-4.1-ESV-R2.tar.gz
```

gcc-c++ライブラリをインストールします。

```
#> yum -y install gcc gcc-c++
```

ISC-DHCP の解凍先フォルダに移り、インストールを実行します。

```
#> cd dhcp-4.1-ESV-R2
#> ./configure --prefix=/usr/local/
#> make && make install
```

以上の操作により、/usr/local/sbin に dhcpd プログラム本体(dhclient, dhcpd, dhcrelay)、/usr/local/bin に omshell、/usr/local/share に各ドキュメントファイル、/usr/local/etc に dhcclient.conf と dhcpd.conf がインストールされます。

次に、ISC-DHCP の正常インストールのテストを目的に、以下の操作によりリースファイルを作成します。

```
#> touch /var/db/dhcpd.leases
```

同様の目的で、設定ファイルである/usr/local/etc/dhcpd.conf を編集します。

設定ファイルの編集後、dhcpd を起動し、正常動作を確認します。

3.3.2. TFTP

ここでは、atftp-server を rpm パッケージからインストールし、xinetd スーパーサーバー上で運用する方法を説明します。
ダウンロードサイ（http://rpmfind.net/linux/rpm2html/search.php?query=atftp-server など）より、rpm パッケージをダウンロードし、次のコマンドでインストールします。

```
#> rpm -ivh atftp-server-x.x.x.rpm
```

ファイルを格納するディレクトリを作ります。

```
#> mkdir /var/lib/tftpboot
#> chown nobody:nobody /var/lib/tftpboot
```

ログ出力用のファイルを作成します。

```
#> touch /var/log/atftp.log
#> chown nobody:nobody /var/log/atftp.log
```

nobody は、atftpd がファイルを操作する際に設定するデフォルトユーザーID です。
このため、/var/lib/tftpboot 及び/var/log/atftp.log の所有者は nobody にします。

/etc/xinetd.d/tftp ファイルを編集します。
※xinetd がインストールされていない場合、# yum -y install xinetd します。

```
service tftp
{
        id = tftp-udp
        disable = no
        socket_type = dgram
        protocol = udp
        wait = yes
        user = root
        #nice = 0
        server = /usr/sbin/atftpd
       server_args = --user nobody.nobody --logfile /var/log/atftp.log --tftpd-timeout 300 --retry-timeout 5 --maxthread 100 --verbose=7 /var/lib/tftpboot
        log_type = SYSLOG local5 info
        flags = IPv4 #デフォルトの IPv6 から変更要。
}
```

xinetd をリロードします。

```
#> /etc/rc.d/init.d/xinetd restart
```

対象となる CM 設定ファイルを/var/lib/tftpboot に配置します。

3.3.3. NTP

Linux ディストリビューション標準の ntpd を使用します。

可用性を高めるために、現用・予備の双方で、外部の複数の上位 NTP と各々独立して同期します。

/etc/ntp.conf を編集します。

〈編集例〉

```
driftfile /var/lib/ntp/drift # ntpd が時刻の変動を記録する driftfile の指定
server ntp1.jst.mfeed.ad.jp # 時刻を同期する上位の NTP サーバー指定
server ntp2.jst.mfeed.ad.jp
```

編集後、ntpd を起動します。

```
#> chkconfig --level 5 ntpd on
#> chkconfig --list ntpd
ntpd      0:off  1:off  2:off  3:off  4:off  5:on  6:off

#> service ntpd start
```

起動後、動作状態を確認します。

```
# ntpq -p
```

接続状況は、行頭の文字で確認できます。主なものは以下の通りです。

* 同期中

+ いつでも同期可能

x クロックが不正確なため無効

空白(スペース) … 以下のいずれかの理由で使用不可

- 不正経路やサーバーダウン等の理由で NTP サーバーと通信不能
- NTP サーバーや経路上のファイアウォール等で UDP123 の受信を拒否
- NTP サーバーが自分自身(ループ)
- 同期処理が進行中

3.3.4. Time

RFC868 Time は、DOCS 端末のイベントログで、ログ表示用時刻として使われます。

DOCS 端末は DHCP オプションから Time サーバーの IP を取得後、時刻情報を取得します。Time からの時刻情報が取得できない場合、DOCS 端末のイベントログに時刻が表示されなくなりますが、DOCS 端末の通信動作には支障ありません。

多くの CMTS には、Time サーバーが標準で付属しています。また、一般的な Linux ディストリビューションにも、同様に Time サーバーが標準で付属しています。

Linux ディストリビューションの Time サーバーは、xinetd 経由で動作するので応答が遅いため、可能な限り CMTS の Time サーバーを使用して下さい。

Linux 付属の Time サーバーを使う場合、以下のように設定します。

```
#> vi /etc/xinetd.d/time-dgram #udp でなく tcp を利用する場合は time-stream を編集。
```

で設定ファイルを編集し、

```
# This is for quick on or off of the service
        disable         = no
```

サービスが有効（disable = no）になっている事を確認します。

※xinetd がインストールされていない場合、# yum -y install xinetd します。

設定ファイルを変更した場合、次のコマンドで xinetd を再起動します。

```
#> /etc/init.d/xinetd restart
```

netstat コマンドで Time サーバーの動作状態を確認します。

```
#> netstat -antu
  Active Internet connections (servers and established)
  Proto Recv-Q Send-Q Local Address Foreign Address State
      tcp  0  0 0.0.0.0:111  0.0.0.0:*  LISTEN
      tcp  0  0 0.0.0.0:22   0.0.0.0:*
      udp  0  0 :::37        :::*       LISTEN
```

37 番ポートが LISTEN していれば動作中です。

3.3.5. DRBD

DRBD をインストールするには、開発元の Linbit 社の以下のサイトからソースをダウンロードし、使用する Linux 環境に合わせた rpm を作成します。

http://oss.linbit.com/drbd/

最初に、rpm を作成する前準備を行います。

```
#> yum -y install gcc make automake autoconf flex rpm-build kernel-devel
#> cd /root
#> mkdir -p rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}
```

DRBD のソースを取得・解凍します。ここでは 8.4.1 の例を示します。
インターネットに接続可能な環境で実行して下さい。

```
#> wget http://oss.linbit.com/drbd/8.4/drbd-8.4.1.tar.gz
#> tar zxvf drbd-8.4.1.tar.gz
```

DRBD フォルダに移り、rpm を作成します。

```
#> cd drbd-8.4.1
#> ./configure
#> make rpm
#> make km-rpm
```

作成した rpm パッケージを現用・予備の両方にインストールします。
オプション U を指定し、既存パッケージがあれば置換します。

```
#> cd /root/rpmbuild/RPMS/x86_64
#> rpm -Uvh drbd-utils-8.4.1-1.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:drbd-utils             ########################################### [100%]
#> rpm -Uvh drbd-km-2.6.32_220.4.1.el6.x86_64-8.4.1-1.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:drbd-km                ########################################### [100%]
#> rpm -Uvh drbd-pacemaker-8.4.1-1.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1: drbd-pacemaker        ########################################### [100%]
```

以上でインストール完了です。

なお Cent OS 5.x の場合、以下のサイトからビルド済の rpm パッケージをダウンロードできますので、こちらを使っても良いでしょう。

http://mirror.centos.org/centos/5/extras/x86_64/RPMS/

DRBD を設定します。現用・予備に同じ設定を行って下さい。

既存設定ファイルを編集します。

```
#> vi /etc/drbd.d/global_common.conf
disk {
   on-io-error detach; #追記 ( IO エラー時にディスクを切り離し)
syncer {
   rate 100M; #追記 ( 同期の帯域幅 )
}
```

新規に設定ファイルを作成・追加します。

```
#> vi /etc/drbd.d/r0.res
resource r0 {
      protocol C ;
   device /dev/drbd0; # DRBD デバイス
      disk /dev/vg_r0/lv_r0; # 物理デバイス
      meta-disk internal;
      on prov01.ossbn.co.jp {
         address 192.168.0.1:7788; # プライマリ DRBD の IP とポート
      }
      on prov02.ossbn.co.jp {
         address 192.168.0.2:7788; # セカンダリ DRBD の IP とポート
      }
}
```

モジュールをロードします。

```
#> modprobe drbd
```

DRBD リソースを作成します。
すべて [yes] か [Enter] を押して進めます。

```
#> drbdadm create-md r0
```

新しいメタデータブロックが作成され、アクティビティログが初期化されます。

現用・予備の双方で DRBD を開始します。

```
#> /etc/rc.d/init.d/drbd start
```

この段階では相手側が未起動なので、以下の問い合わせに yes を入力します。

```
To abort waiting enter 'yes' [ 18]:yes
```

```
#> chkconfig drbd on
#> echo "/sbin/modprobe drbd" >> /etc/rc.local
```

現用・予備への設定完了直後は、いずれもセカンダリの状態です。

各デバイスを接続します。

```
drbdadm up r0
```

デバイスの初期同期の際に、同期ソースとして選択した 1 つのノードに対してのみ実行します。以下のコマンドを実行します。

```
drbdadm -- --overwrite-data-of-peer primary r0
```

暫く時間が経つと同期が完了します。

次に、現用の DRBD デバイスにファイルシステムを作成します。

```
#> mkfs -t ext3 /dev/drbd0
```

最後に、プライマリ側のみマウントします。

```
#> mount /dev/drbd0 /var/db
```

3.3.6. Pacemaker

CentOS 等の RHEL 互換ディストリビューションには、異なるバージョンの Pacemaker が OS に付属している場合があるので確認します。

```
#> rpm -q pacemaker # pacemaker パッケージのインストール状況確認
pacemaker.x86_64 # pacemaker パッケージがインストール済
```

Pacemaker が存在する場合、既存 Pacemaker をアンインストールします。

```
#> yum remove pacemaker
```

その後、http://sourceforge.jp/projects/linux-ha/より、
pacemaker-1.0.11-1.2.2.el6.x86_64.repo.tar.gz をダウンロードし、/tmp に展開します。

```
#> mv pacemaker-1.0.11-1.2.2.el6.x86_64.repo.tar.gz /tmp
#> cd /tmp
tar zxvf pacemaker-1.0.11-1.2.2.el6.x86_64.repo.tar.gz
```

インストール依存している OS 付属の RPM は、ネットワークからダウンロードされます。

```
#> cd pacemaker-1.0.11-1.2.2.el6.x86_64.repo
#> yum -c pacemaker.repo install pacemaker-1.0.11
#> yum -c pacemaker.repo install corosync.x86_64
#> yum -c pacemaker.repo install heartbeat.x86_64
```

インストール終了後、Heartbeat の設定ファイルを編集します。

```
#> vi /etc/ha.d/ha.cf
pacemaker on
logfacility local3 # SYSLOG 出力設定
debug 0 #デバッグ情報の出力抑制
udpport 694  #使用するポート
keepalive 3 #ハートビートの実行間隔
warntime 9 #ハートビート無応答から警告出力までの時間
deadtime 15 #ハートビート無応答からノードダウン認知までの時間
initdead 120 #起動時の deadtime。起動時間分を考慮
bcast eth1 # 1 本目のインターコネクトのインタフェース（直結時）
bcast eth2 # 2 本目のインターコネクトのインタフェース（直結時）
node prov01.ossbn.co.jp #現用マシンのホスト名
node prov02.ossbn.co.jp #現用マシンのホスト名
auto_failback off #自動フェイルバックオフ
watchdog /dev/watchdog # Linux カーネルの Watchdog（Softdog）を利用
ping 192.168.10.254 # DHCP ポートの GW（CMTS の IP 等）を指定。ipfail 用
```

```
respawn hacluster /usr/lib64/heartbeat/ipfail #ipfail の使用宣言
apiauth ipfail uid=hacluster gid=haclient # ipfail の権限
deadping 15 #ping 無応答の判断時間
```

認証ファイルを作成します。

```
#> vi /etc/ha.d/authkeys
auth 1
1 sha1 prov # 任意の文字列。但し、全ノードに同じ文字列を設定します。
```

認証ファイルのパーミッションを 600 に設定します

```
#> chown root:root /etc/ha.d/authkeys
#> chmod 600 /etc/ha.d/authkeys
```

同様の設定を予備にも行います。
設定の完了後、heartbeat を現用・予備で起動します。

```
#> service heartbeat start
```

Pacemaker の起動確認は、状態表示コマンドの crm_mon コマンドを使用します。
Pacemaker の起動には、概ね 1 分程度を要します。

```
#> crm_mon
```

次に、Pacemaker のリソースを設定します。ここでは予めテキストファイルを作成し、ファイルから設定内容を一括反映する方法を説明します。
なお現用 PROV への設定は、heartbeat により予備 PROV に自動的に反映されます。

/usr/lib/ocf/resource.d/prov を作成の上、PROV 用のリソースエージェントファイル(Filesystem)を配置し、パーミッションを 755 に設定します。

-

テキストエディタで設定ファイルを作成します。

```
#> vi /etc/ha.d/pmsetting.txt
```

```
#DRBD 設定
primitive drbd_prov ocf:linbit:drbd ¥
        params drbd_resource="r0" drbdconf="/etc/drbd.conf" ¥
        op start interval="0s" timeout="240s" on-fail="restart" ¥
        op monitor interval="11s" timeout="20s" on-fail="restart" ¥
        op monitor interval="10s" timeout="20s" on-fail="restart" role="Master" ¥
        op stop interval="0s" timeout="120s" on-fail="block"
ms ms_drbd_prov drbd_prov ¥
```

```
        meta  master-max="1"  master-node-max="1"  clone-max="2"  clone-node-max="1"
notify="true"

 #NIC 死活監視
 primitive pingd_prov ocf:pacemaker:pingd ¥
        params  name="default_ping_set"  host_list="192.168.20.2"  multiplier="100"
dampen="0" ¥
        op start interval="0s" timeout="90s" on-fail="restart" ¥
        op monitor interval="10s" timeout="20s" on-fail="restart" ¥
        op stop interval="0s" timeout="100s" on-fail="block"
 clone clnPingd_prov pingd_prov

 #DRBD マウント情報
 primitive fs_prov ocf:prov:Filesystem ¥
        params device="/dev/drbd0" directory="/var/db" fstype="ext3" ¥
        op start interval="0s" timeout="1200s" on-fail="restart" ¥
        op monitor interval="10s" timeout="40s" on-fail="restart" ¥
        op stop interval="0s" timeout="600s" on-fail="block"

 #グループ化
 group group_prov fs_prov

 #マスター/スレーブ設定。prov01 がマスター
 location group_prov-location group_prov ¥
        rule 200: #uname eq prov01 ¥
        rule 100: #uname eq prov02 ¥
        rule -INFINITY: defined default_ping_set and default_ping_set lt 100

 #同じく DRBD のマスター/スレーブ設定
 location master-location_prov ms_drbd_prov ¥
        rule 200: #uname eq prov01 ¥
        rule 100: #uname eq prov02 ¥
        rule role=master -INFINITY: defined default_ping_set and default_ping_set lt
100

 #起動順序と関連性の設定。DRBD をマウントしてから PROV を起動
 colocation prov_on_drbd inf: group_prov ms_drbd_prov:Master
 colocation clnPingd_prov-colocation 1000: group_prov clnPingd_prov
 order order_prov_after_drbd inf: ms_drbd_prov:promote group_prov:start
```

PING 先 IP は環境に
応じて変更して下さい。

```
#Pacemaker の基本動作
property $id="cib-bootstrap-options" ¥
        cluster-infrastructure="openais" ¥
        expected-quorum-votes="2" ¥
        no-quorum-policy="ignore" ¥
        stonith-enabled="false" ¥
        startup-fencing="false" ¥
        dc-deadtime="20s"

#フェイルバック動作の防止
rsc_defaults $id="rsc-options" ¥
        resource-stickiness="INFINITY" ¥
        migration-threshold="1"
```

編集の完了後、設定内容を反映します。

```
#> crm configure
crm(live)configure# load replace /etc/ha.d/pmsetting.txt
crm(live)configure# commit
```

第4章　PROV の運用

4.1.　トラブルシューティング

4.1.1.　　確認ポイント

障害時の確認ポイントを図 4.1.1 に示します。

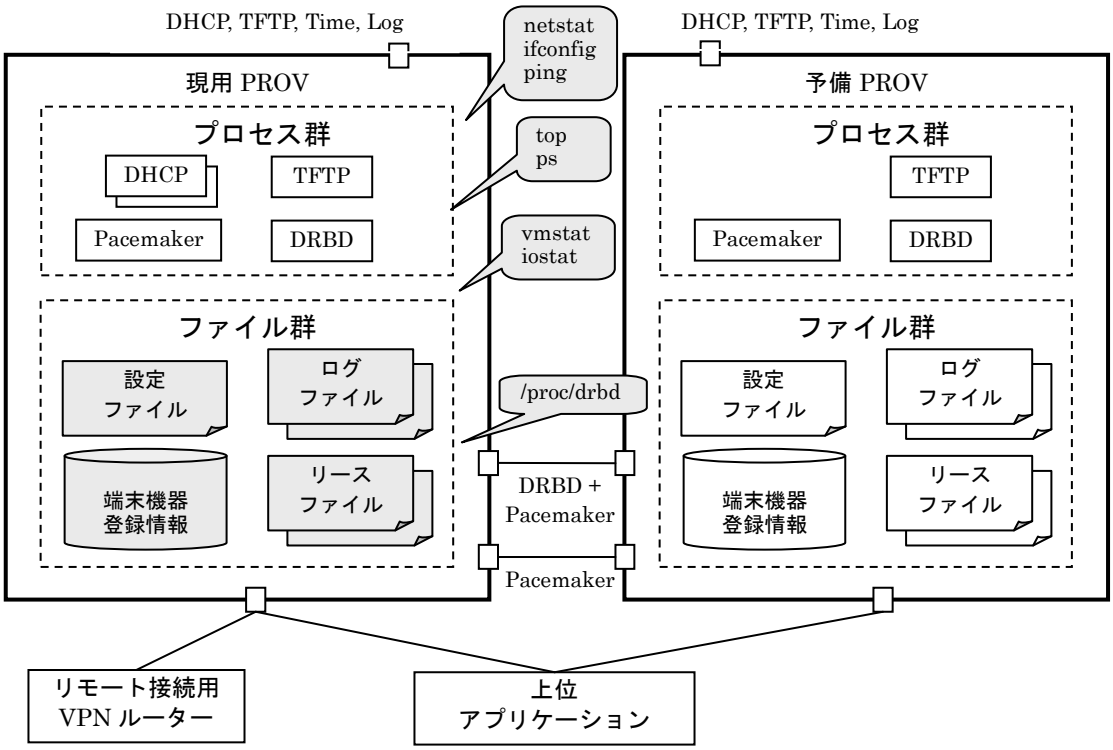


図 4.1.1 障害時の確認ポイント

図中の各確認ポイントを(1)～(5)に示します。

（１）動作状態

動作環境及び各アプリケーションの状態を確認する方法は以下の通りです。

H/W(ハードウェア): LED 点灯色・状態
N/W(ネットワーク): netstat, ifconfig, ping
プロセス: top, ps
I/O: vmstat, iostat
DRBD: /proc/drbd (状態表示用仮想ファイル。cat でファイルの中身を参照)
Pacemaker: crm_mon コマンド
DHCP プロセス ID: /var/run/pid/*.*

（２）動作ログ（ログファイル）

Linux O/S 及び各アプリケーションのログファイルの格納場所は以下の通りです。

O/S, DRBD: /var/log/messages/*.*
Pacemaker: /var/log/pacemaker/*.*
TFTP: /var/log/tftp/*.*
DHCP: /var/log/dhcp/*.*

（３）コア出力先

プロセスクラッシュ時のコアダンプの出力先ファイルは、以下を参照します。

```
/proc/sys/kernel/core_pattern
```

（４）リース情報（リースファイル）

DHCP クライアントへの IP アドレスリース情報の格納場所は以下の通りです。

DHCP プロセスのリースファイル: /var/db/leases/*.*

（５）動作設定（設定ファイル）

各アプリケーションの設定ファイルの格納場所は以下の通りです。

DRBD: /etc/drbd.d/global_common.conf (共通設定), /etc/drbd.conf (個別設定)
Pacemaker: /var/lib/heartbeat/crm/cib.xml (但し編集は crm コマンド経由)
Heartbeat: /etc/ha.d/ha.cf(静的設定), /usr/lib/ocf/resource.d/heartbeat(RA)
TFTP: /etc/xinetd.d/tftp
NTP: /etc/ntp.conf
Time : /etc/inetd.conf, /etc/services
DHCP: /var/db/conf/*.*
Syslog: /etc/syslog.conf, /etc/logrotate.conf

（６）端末機器登録情報

DHCP クライアントの登録情報の格納場所は以下の通りです。

DHCP プロセスとクライアントの関係: /var/db/clients のフォルダとファイル
登録済クライアント: /var/db/clients/*/*.*の名前
定義済グループ・サブクラス: /var/db/clients/*/*.*の内容
DHCP プロセスへの登録内容: /var/db/conf/*.*を tail 参照

4.1.2. 障害分類別の確認ポイント

障害分類別の確認ポイントを表 4.1.2 に示します。

※本書による説明の対象外

障害分類	区分	症状	確認ポイント
DHCP サービス	機能	IP 取得・更新不可	動作状態(N/W, プロセス) 動作ログ(DHCP) リース情報 動作設定(DHCP) 登録情報
		IP 重複	動作ログ(DHCP) リース情報 動作設定(DHCP)
	性能	タイムアウト頻発 取りこぼし 処理時間の間延び	動作状態(N/W, プロセス, I/O) 動作ログ(O/S, DHCP)
TFTP サービス	機能	ファイル取得失敗	動作状態(N/W, プロセス) 動作ログ(TFTP)
		端末起動不可	CM 動作設定ファイル※
	性能	タイムアウト頻発 処理時間の間延び	動作状態(N/W, プロセス, I/O) 動作ログ(O/S, TFTP)
DDNS サービス	機能	更新伝播不可	動作状態(N/W, プロセス) 動作ログ(DHCP) 動作設定(DHCP) 動作設定(DNS)※
上位アプリケーション連携	機能	端末登録不可	動作状態(N/W, プロセス) 動作ログ(上位連携) 動作設定(上位連携) 登録情報
	性能	処理時間の間延び	動作状態(プロセス, I/O)
DRBD 同期処理	機能	同期失敗	動作ログ(DRBD)
		スプリットブレイン	動作ログ(DRBD) 動作ログ(DHCP)
	性能	速度低下	動作状態(DRBD)
Pacemaker フェイルオーバー	機能	切り替え不能	動作ログ(Pacemaker, DRBD) 動作設定(Pacemaker, DRBD)
	性能	処理時間の間延び	動作状態(プロセス, I/O) 動作ログ(Pacemaker, DRBD) 動作設定(Pacemaker, DRBD)
ハードウェア	機能	動作停止	動作状態(H/W, N/W) 動作ログ(O/S) コアダンプ
		カーネルパニック	動作ログ(O/S) コアダンプ
	性能	異常な負荷上昇	動作状態(N/W, プロセス, I/O)
	その他	異臭・異音	動作状態(H/W) 他

表 4.1.2 障害分類別の確認ポイント

4.1.3. フェイルバック操作

ここでは、何らかの障害によりフェイルオーバーが発生した場合の復旧方法について説明します。

まず、どちらか生きているサーバーで crm_mon コマンドを実行し、状況を確認します。
図 5.4.3 は現用 (prov01)が停止し、予備 (prov02)がプライマリとなった場合の表示内容です。
Failed actions:に障害が発生したリソースが表示されます。

```
============
Last updated: Wed Jan 11 01:00:51 2012
Stack: Heartbeat
Current DC: prov02 (f0deb869-6efd-5217-a899-488208a1098f) - partition with quorum
Version: 1.0.11-1554a83db0d3c3e546cfd3aaff6af1184f79ee87
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ prov02 ]
OFFLINE: [ prov01 ]

 Resource Group: group_prov
     fs_prov    (ocf::heartbeat:Filesystem):    Started prov02
 Master/Slave Set: ms_drbd_prov
     Masters: [ prov02 ]
     Stopped: [ drbd_prov:0 ]

Failed actions:
    fs_prov (node=pm01, call=76, rc=7, status=complete): unknown error
```

図 5.4.3 現用サーバー停止時の crm_mon 表示例

尚、crm_mon コマンドを実行後、図 5.4.3 の表示が一向に表示されない場合は、Heartbeat か DRBD が停止している可能性があります。その場合は

```
#> /etc/init.d/heartbeat start
```

及び

```
#> drbdadm up all
```

を実行した後で再度 crm_mon コマンドを実行してください。

次に原因となったサーバー障害を復旧させ、その後、DRBD のデータ同期を復旧させます。DRBD のデータ同期の復旧方法は、障害状況によって異なりますので、詳しくは、以下のページのマニュアルを参考に行ってください。

http://www.drbd.jp/users-guide/ch-troubleshooting.html

ここでは、一番単純な片方のサーバーが電源断等で同期が切れてしまった場合の復旧方法を説明します。まず、プライマリサーバーで DRBD のデータ同期状態を調べます。これには以下を実行します。

#> cat /proc/drbd

図 5.4.4 は、上記の実行時の表示例です。

```
#> cat /proc/drbd
version: 8.4.1 (api:1/proto:86-100)
GIT-hash: 91b4c048c1a0e06777b5f65d312b38d47abaea80 build by root@02, 2011-12-30 16:39:55
 0: cs:StandAlone ro:Primary/Unknown ds:UpToDate/DUnknown   r-----
    ns:0 nr:12 dw:2224 dr:1089 al:34 bm:2 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:2208
```

図 5.4.4 現用サーバー停止時の cat /proc/drbd 表示例

図 5.4.4 で、cs:が接続状態、ro:が自身と相手サーバーの状態を示します。正常時、これらは接続状態が Connected で、自身と相手サーバーの状態が Secondary/Primary 又 Primary/Secondary となります。

接続状態が StandAlone や WFConnection で、自身と相手サーバーの状態が Secondary/Secondary、Primary/Unknown、Secondary/Unknown の場合は、プライマリサーバーで以下を実行することで正常な接続状態に戻ります。

#> drbdadm connect all

実行後は、再度、#> cat /proc/drbd を実行し、状態を再確認してください。復旧しない場合は、前述のマニュアルを参照し対処してください。

障害復旧後、crm_mon コマンドを実行し、Failed actions:が表示されている場合には、各リソースで発生した障害内容をクリアする以下のコマンドを実行してください。

#> crm resource cleanup fs_prov

#> crm resource cleanup ms_drbd_prov

実行後、再度、crm_mon コマンドで Failed actions:が消えているか確認して下さい。これらが消えていないとフェイルバックが出来ません。消えていない場合は、サーバー障害を再確認の上、上記操作を Failed actions:が消えるまで行ってください。

次に、プライマリを予備から現用に戻すには以下のコマンドを実行します。

#> crm resource move fs_prov prov01 force

実行後、移動元のノードにスコア値(-INFINITY)を追加された旨の警告が表示されますので以下を実行します。

#> crm resource unmove fs_prov

実行後は、再度 crm_mon コマンドで両サーバーの状態を確認して下さい。

4.2. ユーティリティーと拡張機能

4.2.1. DHCP 稼動状況の集計出力

リースファイルを集計後、設定ファイルと照合し、現在の貸出状況を出力します。
オープンソース製品例の URL を以下に示します。

C で書かれたツールで、DHCP 設定ファイルとリースファイルを対比し、空き状況を分析し、テキスト形式で表示します。リース履歴が 10 万程度の場合、他ツールが起動から表示までに十数秒を要するのに比べ、本ツールは 3 秒以内で分析を完了するなど高速に動作します。起動時引数として DHCP 設定ファイルとリースファイルのパスを指定できるため、DHCP を定常的に監視するのに便利です。
http://dhcpd-pools.sourceforge.net/

Java で書かれた IP 貸出状況チェック用の GUI アプリケーションです。起動時引数としてリースファイルのパスを指定できるので、DHCP のリース状況を手軽に視認するのに便利です。
http://www.novell.com/coolsolutions/tools/19145.html

Perl と CGI で書かれたツールで、DHCP 設定ファイルとリースファイルを対比し、空き状況やリース一覧を HTML 形式で表示します。DHCP 設定ファイルとリースファイルのパスを自身の設定ファイルに静的に指定します。
http://dhcpstatus.sourceforge.net/
http://www.server-memo.net/server-setting/dhcp/dhcpstatus

Net-SNMP エージェントの拡張モジュールで、DHCP 設定ファイルとリースファイルを対比し、空き状況を分析し、結果を Cacti にグラフ表示します。DHCP 設定ファイルとリースファイルのパスを自身の設定ファイルに静的に指定します。
http://www.net-track.ch/opensource/dhcpd-snmp/

4.2.2. DHCP・TFTP トラフィックの解析

ネットワークトラフィックのキャプチャー・解析には、オープンソース製品のネットワークプロトコルアナライザーである Wire Shark が便利です。
Wire Shark の詳細については、Wire Shark のサイトを参照して下さい。

第5章　外部連携

5.1.　上位アプリケーションとの連携

プリプロビジョニング運用時、上位アプリケーションから DHCP に対し、端末機器 MAC アドレスの一括登録（入庫）、属性情報の変更（契約変更）、MAC アドレスの除外（故障・廃棄）等の操作が発生します。

動作中の DHCP サーバーに MAC アドレスの追加や属性情報の変更を行いたい場合、CLI(OMAPI/omshell)を介して DHCP サーバープロセスのメモリキャッシュとリースファイル(dhcpd.leases)の双方を追記更新するか、ないしは、テキストエディタで設定ファイル(dhcpd.conf)を編集後、対象の DHCP サーバーを停止・再起動します。

再起動は DHCP サービスの可用性を低下させるので、機能上の支障のない限り、設定編集＆再起動よりも、OMAPI を使う方が合理的です。

OMAPI を介してリースファイルに追記更新された設定内容は、操作の度に行が追加されます。削除行は delete フラグが付与される形で残り、DHCP サーバーの再起動時にリースファイルから実際に削除されます。

5.2.　ALA との連携

PROV リファレンスモデル運用時、端末機器に転送される設定ファイル中の ALA の IP アドレス情報により、端末機器とヘッドエンド間の認証・暗号化シーケンスが ALA に誘導されます。

端末機器は設定ファイルの記述に従い、ALA との認証・暗号化セッションを開始します。

ALA セッションは、原理的には ALA と端末機器の双方から開始できますが、SSL/TLS の標準仕様では要求元が PRNG（擬似乱数発生器）により乱数を生成する必要があるため、端末機器の CPU 能力が限られている場合、端末をクライアントとした ALA セッションは、端末機器側の負荷を増大させます。このため ALA 標準では、ALA をクライアントとしたセッションシーケンスを標準としています。この場合の端末機器からの ALA セッション開始要求通知には、Syslog が使われます。

ALA セッションの終了後、ALA は MDM などの端末管理アプリケーションに対し、認証・暗号化の終了を通知します。端末管理システムへの通知には、Syslog が使われます。

5.3. ACS との連携

DHCP オプションによる ACS-URL 通知を PROV に設定したい場合、DHCPv4 options 43 (Vendor Specific Information) と 60 (Vendor Class Identifier)を使用します。

Option 60 には、“dslforum.org”文字列を、記述子の任意の箇所に含めます。

Option 43 には、以下に示すルールにより、文字列形式で各 Parameter を格納します。

Encapsulated Option	Encapsulated Vendor-Specific Option number	Parameter[2]
URL of the ACS	1	ManagementServer.URL
Provisioning code	2	DeviceInfo.ProvisioningCode
CWMP retry minimum wait interval	3	ManagementServer.CWMPRetryMinimumWaitInterval
CWMP retry interval multiplier	4	ManagementServer.CWMPRetryIntervalMultiplier

付録A　DHCP フレームの構造

DHCP メッセージタイプ			
値	メッセージ名	メッセージ送信側	説明
1	DHCP DISCOVER	クライアント	DHCP サーバーを探すためのメッセージ
2	DHCP OFFER	サーバー	DHCP クライアントへの IP 設定情報の候補を通知するメッセージ
3	DHCP REQUEST	クライアント	DHCP サーバーへの IP 設定情報の取得要求メッセージ
4	DHCP DECLINE	クライアント	DHCP サーバーへの IP 設定情報の拒否メッセージ
5	DHCP ACK	サーバー	DHCP クライアントへの IP 設定情報の提供メッセージ
6	DHCP NAK	サーバー	DHCP クライアントへの IP 設定情報の提供拒否メッセージ
7	DHCP RELEASE	クライアント	DHCP サーバーへの IP 設定情報のリリース要求メッセージ
8	DHCP INFORM	クライアント	DHCP サーバーへの IP 以外の設定情報の要求メッセージ

付録B　ISC DHCPv4 オプション

Tag	Name	option-name	option-value type
1	Subnet Mask	subnet-mask	IP address formatted mask
2	Time Offset	time-offset	integer - seconds offset from UTC
3	Router (Default Gateway)	routers	IP address(es) in preferred order
4	Time Server	time-servers	IP address(es) in preferred order
5	Name Server	ien116-name-servers	IP address(es) of IEN 116 name server(s) in preferred order
6	Domain Server	domain-name-servers	IP address(es) in preferred order
7	Log Server	log-servers	IP address(es) in preferred order
8	Quotes Server	cookie-servers	IP address(es) in preferred order
9	LPR Server	lpr-servers	IP address(es) in preferred order
10	Impress Server	impress-servers	IP address(es) in preferred order
11	RLP Server	resource-location-servers	IP address(es) in preferred order
12	Hostname	host-name	text string
13	Boot File Size	boot-size	integer
14	Merit Dump File	merit-dump	text (filename)
15	Domain Name	domain-name	text – client's domain name of the client
16	Swap Server	swap-server	IP address
17	Root Path	root-path	text (directory path)
18	Extension File	extensions-path	text (filename)
19	Forward On/Off	ip-forwarding	* true = enable IP forwarding * false = disable
20	Source Routing	non-local-source-routing	* true = enable forwarding of datagrams with non-local source routing * false = disable
21	Policy Filter	policy-filter	IP address pair(s) consisting of destination and corresponding mask for filtering of source-routed datagrams separated by a space; multiple pairs are comma separated.
22	Max Datagram Size for Reassembly	max-dgram-reassembly	integer (minimum = 576)
23	Default IP TTL	default-ip-ttl	integer
24	MTU Timeout	path-mtu-aging-timeout	integer (seconds)
25	MTU Plateau	path-mtu-plateau-table	integer(s) – (68 is the minimum value of each)
26	MTU Interface	interface-mtu	MTU for a given interface (68 is the minimum value)
27	MTU Subnet	all-subnets-local	true or false
28	Broadcast Address	broadcast-address	IP address

Tag	Name	option-name	option-value type
29	Mask Discovery	perform-mask-discovery	* true = client should perform subnet mask discovery using ICMP * false = client should not do so
30	Mask Supplier	mask-supplier	*true = client should respond to ICMP subnet mask requests * false = client shouldn't respond
31	Router Discovery	router-discovery	* true = client should perform router discovery per RFC 1256 *false = client should not perform router discovery
32	Router Request	router-solicitation-address	IP address
33	Static Route	static-routes	IP address pair(s): destination address and corresponding router separated by space; multiple pairs comma separated; use routers option (3) to specify a default route
34	Trailers	trailer-encapsulation	* true = use trailers (RFC 893) when using ARP * false = do not use trailers
35	ARP Timeout	arp-cache-timeout	integer (seconds)
36	Ethernet	ieee802-3-encapsulation	* true = 802.3 encapsulation * false = Ethernet2 (RFC 894)
37	Default TCP TTL	default-tcp-ttl	integer (seconds)
38	Keepalive Time	tcp-keepalive-interval	integer (seconds)
39	Keepalive Data	tcp-keepalive-garbage	* true = send a garbage octet with keepalive messages for backward TCP compatibility * false = do not send garbage octet
40	NIS Domain	nis-domain	text
41	NIS Servers	nis-servers	IP address(es) in preferred order
42	NTP Servers	ntp-servers	IP address(es) in preferred order
43	Vendor Specific	vendor-encapsulated-options	string of vendor-specific information
44	NETBIOS Name Server	netbios-name-servers	IP address(es) in preferred order
45	NETBIOS Dist Server	netbios-dd-server	IP address(es) in preferred order
46	NETBIOS Node Type	netbios-node-type	integer encoding of node type: 1 = B-node – broadcast no WINS 2 = P-node – WINS only 3 = M-node – broadcast then WINS 4 = H-node – WINS then broadcast
47	NETBIOS Scope	netbios-scope	string encoded in accordance with RFCs 1001-1002
48	X Window Font	font-servers	IP address(es) in preferred order
49	X Window Manager	x-display-manager	IP address(es) in preferred order
50	Address Request	dhcp-requested-address	N/A – used by the client to request a particular IP address
51	Address Time	dhcp-lease-time	N/A – used by client to request a lease time (this option is not configurable on the ISC DHCP server

Tag	Name	option-name	option-value type
			– config file parameters max-lease-time and default-lease-time are used)
52	Overload	dhcp-option-overload	integer indicating which DHCP header field(s) hold options: 1 = “file” field 2 = “sname” field 3 = both “file” and “sname” fields
53	DHCP Message Type	dhcp-message-type	N/A, used to identify message type 1 = DHCPDISCOVER 2 = DHCPOFFER 3 = DHCPREQUEST 4 = DHCPDECLINE 5 = DHCPACK 6 = DHCPNAK 7 = DHCPRELEASE 8 = DHCPINFORM 9 = DHCPFORCERENEW 10=DHCPLEASEQUERY 11=DHCPLEASEUNASIGNED 12=DHCPLEASEUNKNOWN 13=DHCPLEASEACTIVE
54	DHCP Server Identifier	dhcp-server-identifier	N/A – server-identifier parameter in the config file is used instead
55	Parameter List	dhcp-parameter-request-list	integer(s) denoting options to provide to the client
56	DHCP Message	dhcp-message	N/A – used to populate an error message from the server with a DHCPNAK or from a client in a DHCPDECLINE message
57	DHCP Max Msg Size	dhcp-max-message-size	integer – used as a default if not provided by the client
58	Renewal Time	dhcp-renewal-time	N/A – based on the lease time
59	Rebinding Time	dhcp-rebinding-time	N/A – based on the lease time
60	Vendor Class Id	vendor-class-identifier	string
61	Client Id	dhcp-client-identifier	text (string or hex digits) – be aware that some clients prepend a 0 to the ASCII text so you may have to encode "¥¥ 0foo" instead of just “foo”
62	Netware/IP Domain	nwip-domain	string
63	Netware/IP Options – specify as single hex string or per suboption	nwip-suboptions nwip.nsq-broadcast nwip.preferred-dss nwip.nearest-nwip-server nwip.autoretries nwip.autoretry-secs nwip.nwip-1-1	string of all options in hex flag IP address(es) IP address(es) integer integer integer

Tag	Name	option-name	option-value type
		nwip.primary-dss	IP address
64	NIS+ Domain Name	nisplus-domain	text (domain name)
65	NIS+ Server Address	nisplus-servers	IP address(es) in preferred order
66	Server Name	tftp-server-name	text (server domain name)
67	Bootfile Name	bootfile-name	text (file name)
68	Home Agent Addresses	mobile-ip-home-agent	IP address(es) in preferred order
69	SMTP Server	smtp-server	IP address(es) in preferred order
70	POP3 Server	pop-server	IP address(es) in preferred order
71	NNTP Server	nntp-server	IP address(es) in preferred order
72	WWW Server	www-server	IP address(es) in preferred order
73	Finger Server	finger-server	IP address(es) in preferred order
74	IRC Server	irc-server	IP address(es) in preferred order
75	StreetTalk Server	streettalk-server	IP address(es) in preferred order
76	StreetTalk Directory Assistance (STDA) Server	streettalk-directory-assistance-server	IP address(es)
77	User Class	user-class	string
78	Service Location Protocol (SLP) Directory Agent	slp-directory-agent	Two parameters (space separated): * Boolean – true = use only addresses provided in this option; false = use provided options and others via SLP agent discovery * IP address(es) of SLP directory agent(s)
79	SLP Service Scope	slp-service-scope	Two parameters (space separated): * Boolean – true = use only service scope(s) provided in this option; false = use provided scope(s) and others statically configured * text string(s) of service scope(es)
80	Rapid Commit		N/A – sent by client to request 2 packettransaction instead of the normal 4 packet transaction
81	Client FQDN	fqdn option space	N/A – sent by client regarding DDNS update. Please refer to the Client FQDN Option page for server parameters associated with this option
82	Relay Agent Information	agent.circuit-id agent.remote-id agent.DOCSIS-device-class agent.link-selection	N/A for setting values. Please refer to the Relay Agent Information Options page for more details.
83	iSNS	*Not natively supported*	N/A
85	NDS Servers	nds-servers	IP address(es)
86	NDS Tree Name	nds-tree-name	string
87	NDS Context	nds-context	string
88	BCMCS Controller Domain Name	bcms-controller-names	domain list
89	BCMCS Controller	bcms-controller-address	IP address list

Tag	Name	option-name	option-value type
	IPv4 address option		
90	Authentication		N/A – used to pass DHCP authentication information
91	Client-last-transaction-time option		N/A – used as part of DHCP Lease Query
92	Associated-ip option		N/A – used as part of DHCP Lease Query
93	Client System		N/A – used by PXE clients to convey the client hardware architecture to the server
94	Client NDI		N/A – used by PXE clients to convey the client's network interface type to the server
95	LDAP	*Not natively supported*	N/A
97	UUID/GUID		N/A – used by PXE clients to convey the client's unique identifier to the server
98	User Authentication Servers	uap-servers	text (URL list)
99	GEOCONF_ CIVIC	*Not natively supported*	N/A
112	Netinfo Address	netinfo-server-address	IP address(es) – not defined in any RFC but assigned by IANA
113	Netinfo Tag	netinfo-server-tag	test string - – not defined in any RFC but assigned by IANA
114	Default URL	default-url	text (URL) – not defined in any RFC but assigned by IANA
116	Auto-Config		N/A – used by clients to request auto-configuration support from the server
117	Name Service Search	*Not natively supported*	N/A
118	Subnet Selection Option	subnet-selection	N/A – client preferred subnet for address assignment
119	Domain Search	domain-search	One or more domain names, each enclosed in quotes and separated by commas
120	SIP Servers DHCP Option	*Not natively supported*	N/A
121	Classless Static Route Option	*Not natively supported*	N/A
122	CCC	*Not natively supported*	N/A
123	GeoConf Option	*Not natively supported*	N/A
124	Vendor Identified Vendor Class		N/A – sent by client to convey its vendor-identified vendor class
125	Vendor Identified Vendor-Specific Information	vivso	string

付録C IEC62056 スマートメーター対応

DHCP の設定

(1) オプションパラメーター

以下の通りマッピングします。

No.	通知内容	DHCP オプション名		データ型	備考
1	TFTP サーバーIP	66	tftp-server-name	text	next-server にも定義
2	NTP サーバーIP	42	ntp-servers	IP address(es) in preferred order	
3	SYSLOG サーバーIP	98	log-servers	IP address(es) in preferred order	

(2) DHCP の構成

標準的な単一の dhcpd.conf、dhcpd.leases で定義された dhcpd プロセスとします。

(3) 動作設定ファイルの編集

動作設定ファイル(/etc/dhcp/dhcpd.conf)を編集します。詳細は以下の通りです。

```
### Top Level ###
authoritative;
ddns-update-style none;
deny bootp;
ping-check false;
deny leasequery;

### Group ###
group sm_test01 {
                max-lease-time 86400;
                default-lease-time 86400;
                option tftp-server-name "192.168.101.103";
                next-server 192.168.101.103;
                option ntp-servers 192.168.101.103;
                option log-servers 192.168.101.103;
                filename "sm_test01.zip";
                option bootfile-name "sm_test01.zip";
}

### Subnet ###
subnet 192.168.101.0 netmask 255.255.255.0 {
}

### Host ###
host 0026b988d7a6 {
                hardware ethernet 00:26:b9:88:d7:a6;
                group "sm_test01";
                fixed-address 192.168.101.201;
}

host 5453ed1d32a9 {
                hardware ethernet 54:53:ed:1d:32:a9;
                group "sm_test01";
                fixed-address 192.168.101.202;
}
```

"group"にメータ機種別に起動パラメータを定義。
・next-server　… TFTP サーバアドレス
※option tftp-server-name も上記と同様だが念のため追加。
・option ntp-server … NTP サーバアドレス
・option log-servers … Syslog サーバアドレス
・filename　… イメージファイル名
※option bootfile-name も上記と同様だが念のため追加。

subnet は nic の IP アドレスに合わせて定義。
※中身は何もなくても問題なし。

スマートメータシミュレータの MAC アドレスと、対応する"group"と割り当てる固定 IP アドレスを定義。

host 名はユニークであれば何でも構わないがここでは MAC アドレスを指定。

「/etc/dhcp/dhcpd.conf」の設定例

TFTP の設定

(1) スレッド構成

マルチスレッド構成とします。

TFTP は OS 標準のもので無く、マルチスレッドに対応した”atftp”を使用します。

rpm パッケージは、以下のサイトからダウンロード可能です。

http://rpmfind.net/linux/rpm2html/search.php?query=atftp-server

以下のコマンドでインストールします。

#> rpm -ivh atftp-server-x.x.x.rpm　　※x.x.x はバージョン番号

ファイルを格納するディレクトリを作成します。

#> mkdir /var/lib/tftpboot

#> chown nobody:nobody /var/lib/tftpboot

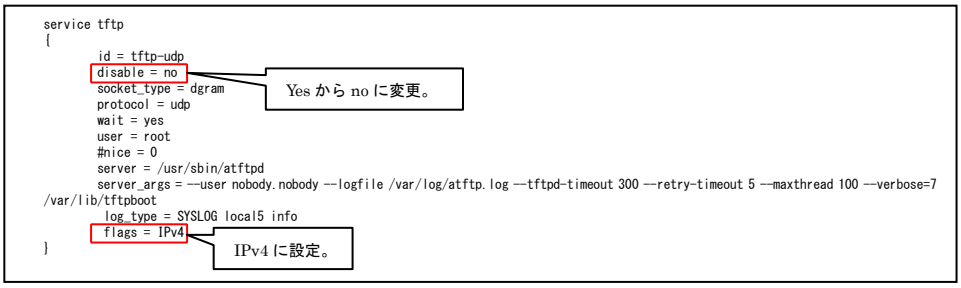
ログ出力用のファイルを作成します。

#> touch /var/log/atftp.log

#> chown nobody:nobody /var/log/atftp.log

nobody は、atftpd がファイルを操作する際に設定するデフォルトユーザーID です。
このため、/var/lib/tftpboot 及び/var/log/atftp.log の所有者は nobody に設定します。

動作設定ファイル(/etc/xinetd.d/tftp)を編集します。詳細は以下の通りです。



「/etc/xinet.d/tftp」の設定例

最後に、スーパーサーバを再起動します。

#> service xinetd restart

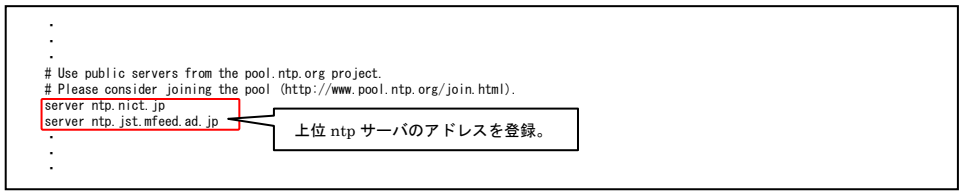
NTP の設定

(1) 動作モード

動作モードには、slew を選択します。

slew は、時刻の進み具合をずらしていくことで、少しずつ時刻を合わせていく方式です。

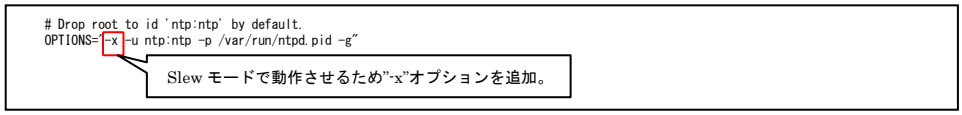
動作設定ファイル(/etc/ntp.conf)を編集します。



「/etc/ntp.conf」の編集例　※編集箇所以外は省略

上図は設定例です。実際の環境に合わせ、適宜調整を行って下さい。

次に、起動設定ファイル(/etc/sysconfig/ntpd)を編集します。



「/etc/sysconfig/ntpd」の編集例

編集後、ntpd を起動します。

```
#> service ntpd start
```

OPEN LIB PROV エンジニアリングガイド

2014 年 5 月 7 日 第 0.6 版 発行

著　　者　　宮副　英治、中武　正文、笹沼　元秀、西出　誠
発　　行　　オーエスエスブロードネット株式会社
〒 213-0011 神奈川県川崎市高津区久本 3-5-7 新溝ノロビル 5F
電子メール: info@ossbn.co.jp
