



## DOCSIS 設定

---

この章では、Broadband Access Center (BAC) DOCSIS 配備のプロビジョニング フローについて説明します。また、設定に先立って必要な情報を説明し、使用可能なツールについて説明します。

- [DOCSIS ワークフロー \(P.6-2\)](#)
- [動的 DOCSIS テンプレートによる MIB の使用方法 \(P.6-9\)](#)
- [DOCSIS 設定のための BAC 機能 \(P.6-10\)](#)
- [IPv6 のサポート \(P.6-13\)](#)
- [リース クエリー \(P.6-19\)](#)



(注)

---

この BAC リリースでサポートされている DOCSIS オプションについては、[P.B-2 の「DOCSIS オプションのサポート」](#)を参照してください。

---

この章は、ユーザが次の仕様の内容を熟知していることを想定しています。

- DOCSIS 3.0 :
  - CM-SP-SECv3.0-I04-070518
  - CM-SP-PHY3.0-I04-070518
  - CM-SP-MULPIv3.0-I04-070518
  - CM-SP-OSSIV3.0-I03-070518
- DOCSIS 2.0 :
  - CM-SP-RFI2.0-I12-071206
  - L2VPN CM-SP-L2VPN-I06-071206

## DOCSIS ワークフロー

この項では、DHCPv4 と DHCPv6 の DOCSIS プロビジョニング仕様に含まれるプロビジョニングワークフローについて説明します。

- [DOCSIS DHCPv4 ワークフロー \(P.6-2\)](#)
- [DOCSIS DHCPv6 ワークフロー \(P.6-5\)](#)

## DOCSIS DHCPv4 ワークフロー

図 6-1 に、DHCPv4 の DOCSIS プロビジョニング仕様に含まれるプロビジョニングワークフローを示します。その後、各ステップについて説明します。

図 6-1 DOCSIS DHCPv4 プロビジョニングフロー

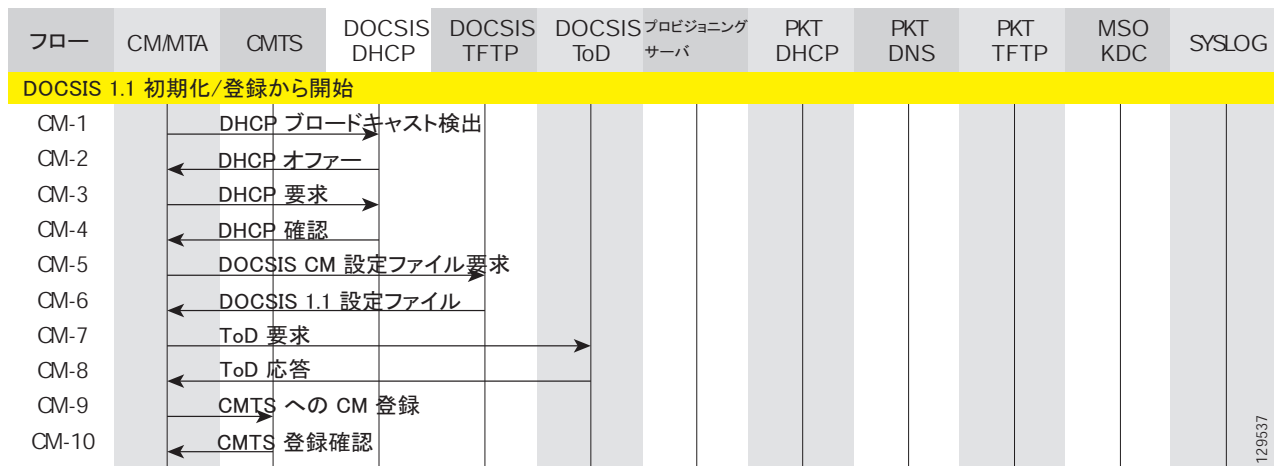


表 6-1 で、図 6-1 に示されるさまざまな DOCSIS プロビジョニング ステップで発生する可能性がある問題について説明します。

表 6-1 DOCSIS DHCPv4 ワークフローの説明

手順	DOCSIS DHCPv4 ワークフロー	潜在的な問題
CM <sup>1</sup> -1	DHCP 検出	<ul style="list-style-type: none"> <li>init(d) 状態。</li> <li>使用可能なアドレスがない。</li> <li>BAC 共有秘密情報に誤りがある。</li> <li>サービス クラスが誤って設定される。</li> <li>DOCSIS テンプレートの解析エラー（無効なオプション、インクルードファイルが見つからないなど）。</li> </ul> <p><b>Cisco Network Registrar DHCP</b></p> <ul style="list-style-type: none"> <li>DHCP 設定に誤りがある。</li> <li>DHCP サーバがプロビジョニング グループ内に存在しない。</li> </ul> <p><b>BAC Network Registrar 拡張</b></p> <ul style="list-style-type: none"> <li>Network Registrar 拡張が DPE に接続できない。</li> <li>Network Registrar 拡張がプロビジョニング グループ内の DPE を検出できない。</li> <li>拡張が RDU に接続されていることを確認する。</li> <li>Network Registrar 拡張が DPE キャッシュ ミスを取得し、要求を RDU に送信する。</li> </ul> <p><b>RDU</b></p> <ul style="list-style-type: none"> <li>適切なスコープが定義されていない(または RDU の設定と一致しない)。</li> <li>RDU の IP アドレスに誤りがある。</li> <li>RDU ポートに誤りがある（デフォルト 49187）。</li> <li>DPE から RDU に PING できない。</li> <li>RDU で構成の生成に失敗している。</li> <li>RDU ライセンス数が超過したため、設定されない。</li> <li>RDU でデバイスの検出に失敗している。</li> </ul> <p><b>DPE</b></p> <ul style="list-style-type: none"> <li>DPE がプロビジョニング グループに割り当てられていない。</li> <li>DHCP サーバから DPE に PING できない。</li> <li>DPE インターフェイスでプロビジョニングがイネーブルになっていない。</li> </ul>
CM-2	DHCP オファー	DHCP と Cable Modem Termination System (CMTS; ケーブル モデム ターミネーション システム) 間のルーティングの問題。
CM-3	DHCP 要求	<ul style="list-style-type: none"> <li>init(i) 状態。</li> <li>DHCP サーバからすべての必須パラメータが提供されなかった。</li> </ul>
CM-4	DHCP 確認	

表 6-1 DOCSIS DHCPv4 ワークフローの説明 (続き)

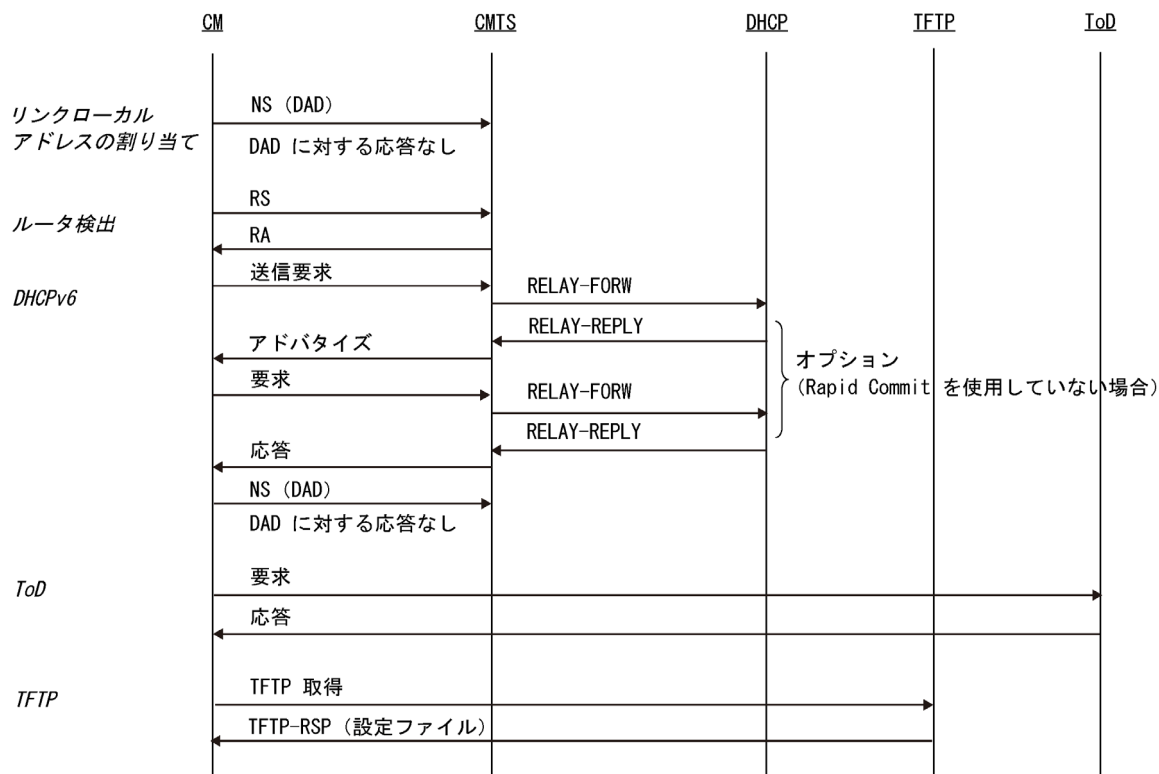
手順	DOCSIS DHCPv4 ワークフロー	潜在的な問題
CM-5	TFTP 要求	<ul style="list-style-type: none"> <li>• Init(o) 状態。</li> <li>• CMTS と DPE 間のルーティングの問題。</li> <li>• TFTP サーバ (DPE) からモデムへのルートがない。</li> <li>• DPE キャッシュ ミス (静的ファイル、RDU がダウンしているかファイルがない)。</li> <li>• TFTP サーバ (DPE) でファイルが見つからない。</li> <li>• DPE キャッシュ ミス (動的ファイル)。</li> <li>• DPE IP 検証エラー (たとえば、デバイスの IP アドレスが予期したアドレスではない、動的共有秘密が CMTS でイネーブルになっている、DOCSIS モデムとしてハッカーがスプーフィングしている)。</li> </ul>
CM-6	TFTP 応答	DPE と CMTS 間のルーティングの問題。
CM-7	ToD 要求	init(t) 状態: タイム サーバ (DPE) からモデムへのルートがない。
CM-8	ToD 応答	
CM-9	CMTS への CM の登録	<ul style="list-style-type: none"> <li>• reject(m) : * CMTS 共有秘密情報が BAC または DPE DOCSIS 共有秘密情報と一致しない。</li> <li>• reject(c) : * 誤った DOCSIS 設定ファイルが配信された (1.1 ファイルを 1.0 ケーブル モデムへ配信)。</li> </ul>
CM-10	CMTS 登録確認応答	許容できる状態 : <ul style="list-style-type: none"> <li>• online</li> <li>• online(d)</li> <li>• online(pk)</li> <li>• online(pt)</li> </ul>

1. CM = ケーブル モデム

## DOCSIS DHCPv6 ワークフロー

図 6-2 に、DHCPv6 の DOCSIS プロビジョニング仕様に含まれるプロビジョニング ワークフローを示します。その後、各ステップについて説明します。

図 6-2 DOCSIS DHCPv6 プロビジョニング フロー



DHCPv6 の DOCSIS プロビジョニング ワークフローには、次の割り当てを含む、ケーブル モデムによる IPv6 接続の確立が含まれます。

- リンクローカルアドレス
- デフォルトルータ
- IPv6 管理アドレス
- その他の IPv6 の設定

表 6-2 で、図 6-2 に示されるさまざまな DOCSIS プロビジョニング ステップで発生する可能性がある問題について説明します。

表 6-2 DOCSIS DHCPv6 ワークフローの説明

ワークフロー	説明	潜在的な問題
<b>プロビジョニング段階：リンクローカルアドレスの割り当て</b>		
	ケーブル モデムは、インターフェイスの MAC アドレスから取得する EUI-64 (64 ビット Extended Unique Identifier) から IPv6 リンクローカルアドレスを構築します。	
NS (DAD)	ケーブル モデムは、ネイバー送信要求 (NS) メッセージを使用して、Duplicate Address Detection (DAD; 重複アドレス検出) を実行します。DAD は、構築したリンクローカルアドレスがすでに使用中であるかどうかを確認します。NS に対する応答がない場合、ケーブル モデムはリンクローカルアドレスが未使用であると判断します。応答が返ってきた場合は、リンクローカルアドレスと MAC アドレスが競合していることを示し、ケーブル モデムはプロビジョニングプロセスを停止します。	
<b>プロビジョニング段階：ルータ検出</b>		
	ケーブル モデムは、ルータ検出を使用してデフォルトのルータを検出し、HFC リンクのプレフィックスを特定します。	
RS	ケーブル モデムは、ルータ送信要求 (RS) を CMTS に送信して、定期的にルータアドバタイズメント (RA) メッセージの転送をトリガーします。	
RA	CMTS ルータは、定期的に RA を送信します。各 RA には次のものが含まれます。 <ul style="list-style-type: none"> <li>リンクに割り当てる IPv6 プレフィックスのリスト</li> <li>DHCPv6 を使用するためのディレクティブ</li> <li>デフォルト ルータとしての CMTS ルータの Availability</li> </ul>	

表 6-2 DOCSIS DHCPv6 ワークフローの説明 (続き)

ワークフロー	説明	潜在的な問題
プロビジョニング段階: DHCPv6		
送信要求	ケーブル モデムは、送信要求メッセージを送信して DHCP サーバを検索します。	<ul style="list-style-type: none"> <li>• init6(s) 状態。</li> <li>• 使用可能な IPv6 アドレスがない。</li> <li>• BAC 共有秘密情報に誤りがある。</li> <li>• サービス クラスが誤って設定される。</li> <li>• DOCSIS テンプレートの解析エラー (無効なオプション、インクルード ファイルが見つからないなど)。</li> </ul> <p><b>Network Registrar DHCP</b></p> <ul style="list-style-type: none"> <li>• DHCPv6 設定に誤りがある。</li> <li>• DHCP サーバがプロビジョニング グループ内に存在しない。</li> <li>• 適切なプレフィックスが定義されていない (または BAC RDU 設定と一致しない)。</li> </ul> <p><b>BAC Network Registrar 拡張</b></p> <ul style="list-style-type: none"> <li>• Network Registrar 拡張が DPE に接続できない。</li> <li>• Network Registrar 拡張がプロビジョニング グループ内の IPv6 DPE を検出できない。</li> <li>• 拡張が RDU に接続されていることを確認する。</li> <li>• Network Registrar 拡張が DPE キャッシュ ミスを取得し、要求を RDU に送信する。</li> </ul> <p><b>RDU</b></p> <ul style="list-style-type: none"> <li>• RDU の IP アドレスに誤りがある。</li> <li>• RDU ポートに誤りがある (デフォルト 49187)。</li> <li>• DPE から RDU に PING できない。</li> <li>• RDU で構成の生成に失敗している。</li> <li>• RDU ライセンス数が超過したため、設定されない。</li> <li>• RDU でデバイスの検出に失敗している。</li> </ul> <p><b>DPE</b></p> <ul style="list-style-type: none"> <li>• DPE がプロビジョニング グループに割り当てられていない。</li> <li>• DHCP サーバから DPE に PING できない。</li> <li>• DPE インターフェイスで IPv6 プロビジョニング がイネーブルになっていない。</li> <li>• IPv6 プロビジョニングのプロビジョニング グループがイネーブルになっていない。</li> </ul>

表 6-2 DOCSIS DHCPv6 ワークフローの説明 (続き)

ワークフロー	説明	潜在的な問題
Relay-Forw	リレー エージェントは、ケーブル モデムから受信する DHCPv6 メッセージ全体を DHCPv6 サーバに転送します。  リレー エージェントは、次のようなリレー エージェントのメッセージ フィールドとオプションを追加します。 <ul style="list-style-type: none"> <li>Peer-address</li> <li>Link-address</li> <li>Interface ID</li> </ul>	
Relay-Repl	リレー エージェントは、サーバの応答を抽出して、CMTS 経由でケーブル モデムに転送します。	
アドバタイズ	DHCP サーバは、ケーブル モデムから受信した送信要求メッセージに対する応答として、DHCP サービスでサーバを利用可能なことを示すアドバタイズメッセージを返します。	<ul style="list-style-type: none"> <li>init6(a) 状態。</li> <li>DHCP と CMTS 間のルーティングの問題。</li> </ul>
要求	アドバタイズ メッセージを受信すると、ケーブル モデムは要求メッセージを送信して、特定のサーバの IP アドレスを含む、構成パラメータを要求します。	<ul style="list-style-type: none"> <li>init6(r) 状態。</li> <li>DHCP サーバからすべての必須パラメータが提供されなかった。</li> </ul>
Relay-Forw	リレー エージェントは、メッセージを DHCPv6 サーバに転送します。	
Relay-Repl	リレー エージェントは、サーバの応答を抽出して、CMTS 経由でケーブル モデムに転送します。	
応答	CMTS は、割り当て済みアドレスと構成パラメータを含む、DHCP サーバから受信した応答メッセージを転送します。	init6(i) 状態。



**(注)** DHCPv6 クライアントは、Rapid Commit モードでプロビジョニングできます。Rapid Commit は、通常の 4 つのメッセージ交換の代わりに、2 つのメッセージ交換を行います。2 つのメッセージ交換には、送信要求と応答が含まれます。4 つのメッセージ交換には、送信要求、アドバタイズ、要求、応答が含まれます。これらメッセージはすべて、リレー エージェントを通過する場合、Relay-Forw または Relay-Repl メッセージにラップされます。

Rapid Commit がイネーブルになっている場合は、DHCP サーバは送信要求 (Relay-Forw メッセージにラップ) メッセージに対し 応答 (Relay-Repl メッセージにラップ) メッセージで応答します。Rapid Commit をディセーブルにすると、DHCP サーバはアドバタイズ (Relay-Repl にラップ) メッセージで応答します。

NS (DAD)	DHCPv6 メッセージの交換が完了すると、ケーブル モデムは、リンクローカルアドレスが DAD 経由ですでに使用中であるかどうかを確認します。応答を受信しない場合、ケーブル モデムは IP アドレスの取得に成功したと判断します。	
<b>プロビジョニング段階: ToD</b>		
要求	IPv6 アドレスを取得後、ケーブル モデムは RFC 868 タイム サーバの Time of Day を要求します。	init6(t) 状態: タイム サーバ (DPE) からモデムへのルートがない。
応答	サーバの IPv6 アドレスは、DHCPv6 オプションを使用して指定します。	



表 6-2 DOCSIS DHCPv6 ワークフローの説明（続き）

ワークフロー	説明	潜在的な問題
プロビジョニング段階：TFTP		
TFTP 取得	ケーブル モデムは、TFTP を使用して設定ファイルをダウンロードします。サーバの IPv6 アドレスと設定ファイルの名前は、DHCPv6 経由で利用可能になります。	<ul style="list-style-type: none"> <li>• init6(o) 状態。</li> <li>• CMTS と DPE 間のルーティングの問題。</li> <li>• TFTP サーバ(DPE)からモデムへのルートがない。</li> <li>• DPE キャッシュ ミス（静的ファイル、RDU がダウンロードしているかファイルがない）。</li> <li>• TFTP サーバ（DPE）でファイルが見つからない。</li> <li>• DPE キャッシュ ミス（動的ファイル）。</li> <li>• DPE IP 検証エラー（たとえば、デバイスの IP アドレスが予期したアドレスではない、Dynamic Shared Secret が CMTS でイネーブルになっている、DOCSIS モデムとしてハッカーがスプーフィングしている）。</li> </ul>
TFTP RSP（設定ファイル）		DPE と CMTS 間のルーティングの問題。
これで、ケーブル モデムが IPv6 操作用にプロビジョニングされました。		

## 動的 DOCSIS テンプレートによる MIB の使用方法

BAC に付属する MIB の完全なリストについては、[P.5-7](#) の「[SNMP VarBind](#)」を参照してください。

RDU にロードされる DOCSIS MIB には、2つのバージョンがあります。

- DOCS-CABLE-DEVICE-MIB-OBSOLETE（実験的ブランチ）
- DOCS-CABLE-DEVICE-MIB（mib2 ブランチ）

これらの使用方法については、[P.5-7](#) の「[DOCSIS MIB](#)」を参照してください。

Application Programming Interface（API; アプリケーションプログラミングインターフェイス）コールを使用するか、*rdu.properties* を修正することにより、MIB を追加することができます。詳細については、[P.7-34](#) の「[Euro-PacketCable MIB の設定](#)」を参照してください。

次の場合に、SNMP TLV をテンプレートに追加できます。

- 利用可能な MIB がないとき。[P.5-11](#) の「[MIB を使用しない SNMP TLV の追加](#)」を参照してください。
- ベンダー固有の MIB を使用するとき。[P.5-12](#) の「[ベンダー固有の MIB を使用した SNMP TLV の追加](#)」を参照してください。

## DOCSIS 設定のための BAC 機能

この項では、DOCSIS テクノロジーと関連する BAC の付加価値機能について説明します。

### 動的設定 TLV

DPE は、動的 DOCSIS 設定の TFTP 要求を受け取るたびに、次の TLV を追加します。

- TLV 19 : TFTP サーバのタイムスタンプ (オプション) : **Configure DOCSIS Defaults** ページに TFTP Time Stamp Option として表示されます。詳細については、表 13-3 を参照してください。この TLV では、CMTS および DPE における NTP の同期が必要です。
- TLV 20 および TLV 59 : IPv4 と IPv6 用の TFTP サーバのプロビジョニングされたモデムアドレス (オプション) : **Configure DOCSIS Defaults** ページに TFTP Modem Address Option として表示されます。詳細については、表 13-3 を参照してください。



(注) DPE の TFTP IP 検証機能は、Cisco CMTS DSS 機能とは互換性がありません。P.6-10 の「DPE TFTP IP 検証」を参照してください。Cisco CMTS に DSS が設定されている場合、TFTP サーバのプロビジョニングされたモデムアドレスをディセーブルにする必要があります。

- TLV 6 : CM MIC の設定 (必須)
- TLV 7 : CMTS MIC の設定 (必須) : **Configure DOCSIS Defaults** ページに CMTS Shared Secret として表示されます。詳細については、表 13-3 を参照してください。
- TLV 43.6.x : 拡張 CMTS MIC の設定 (必須) : **Configure DOCSIS Defaults** ページに CMTS Shared Secret として表示されます。詳細については、表 13-3 を参照してください。



(注) CMTS MIC を設定するときは、次の CMTS IOS リリースの依存関係に注意してください。

- TLV 39 または TLV 40 を含める場合、DOCSIS 2.0 CMTS MIC には CMTS IOS 12.3BC が必要です。
- 次の CMTS IOS コマンドが BAC によって設定されているものとします。
  - **ip dhcp relay information option**
  - **no ip dhcp relay information check**
  - **cable helper-address x.x.x.x**

ここで、x.x.x.x は Network Registrar DHCP サーバの IP アドレスです。

IPv6 環境では、**cable helper-address** の代わりに次のコマンドを使用する必要があります。

  - **ipv6 dhcp relay destination ipv6-address [interface-type interface-number]**
  - **cable dhcp-giaddr primary**

### DPE TFTP IP 検証

DPE TFTP サーバは、動的設定ファイルを対象に、TFTP クライアントの IP アドレスが DOCSIS ケーブル モデムの予期した IP アドレスと一致することを確認します。一致しない場合、要求は破棄されます。この機能に Cisco CMTS DMIC 機能との互換性はありません。

TFTP の動的構成要求における要求者の IP アドレスの検証をディセーブルにするには、**no service tftp 1..1 ipv4 | ipv6 verify-ip** コマンドを使用します。詳細については、『Cisco Broadband Access Center DPE CLI Reference 4.0』を参照してください。

## DOCSIS 1.0、1.1、2.0、および 3.0 のサポート

BAC 4.0 は、DOCSIS 1.0、1.1、2.0、および 3.0 をサポートします。TLV の詳細については、P.5-2 の「[テンプレート文法](#)」を参照してください。この BAC リリースがサポートする各 DOCSIS バージョンのオプションのリストについては、P.B-2 の「[DOCSIS オプションのサポート](#)」を参照してください。

## DOCSIS バージョンの動的選択

BAC は、ケーブルモデムの DOCSIS バージョンを着信 DHCP 要求から検出できます。また、次の 2 つの方法のいずれかで CMTS の DOCSIS バージョンを検出することもできます。

- DHCPv4 の Option 82 と DHCPv6 の Option 17 を使用して、CMTS をその DOCSIS バージョンを送信するリレー エージェントとして使用する方法。
- GIADDR から CMTS DOCSIS バージョンへのマッピングを行う顧客提供のソースを使用する方法。

この DOCSIS バージョンを使用して、BAC は、デバイスの最適な DOCSIS 設定ファイルを判断します（そのように設定されている場合）。このバージョンが、デバイスと CMTS 間で最も一般的な DOCSIS バージョンです。たとえば、デバイスが DOCSIS 2.0 をサポートし、CMTS が DOCSIS 1.1 をサポートする場合、DOCSIS 1.1 ファイルが使用されます。

### モデムの DOCSIS バージョンの判別

BAC は、ケーブルモデムの DOCSIS バージョンを着信 DHCP 要求から検出できます。この要求のモデムの機能を示す Vendor Class Identifier フィールド (Option 60) に文字列が含まれています。たとえば、「docsis1.1:xxxxxx」では、xxxxxx はモデムの機能を表す ASCII 文字列です。サービスレベル選択拡張は、「docsis」と「:xxxxxx」16 進文字列間の文字をモデムの DOCSIS バージョンとして使用します。

### CMTS の DOCSIS バージョンの判別

この BAC リリースでは、CMTS をイネーブルにしてリレー エージェントとして動作させ、CMTS の DOCSIS バージョンを提供できます。この機能は次のオプションを使用してイネーブルにします。

- DHCPv4 Relay Agent Option 82。このオプションを使用すると、CMTS は CMTS の特定機能を送信（またはアドバタイズ）できます。このオプションは、DOCSIS DHCP ベンダー指定のオプションで、CMTS の DOCSIS バージョンを伝達します。
- DHCPv6 Vendor-specific Information Option 17。このオプションを使用すると、ベンダー固有の情報を指定できます。このオプションは、Relay-forward メッセージと Relay-reply メッセージで伝達され、DHCPv6 リレー エージェントと DHCPv6 サーバ間で情報を送信します。

以前のバージョンと同様、この BAC バージョンは、DHCP GIADDR フィールドを介して CMTS の DOCSIS バージョンを判別します。このフィールドは、CMTS インターフェイスの IP アドレスを指定します。この方法の場合、DOCSIS モデムのサービスレベル選択拡張は、`/docsis/cmts/version/giaddrToVersionMap` プロパティを検索します。このプロパティの値は、GIADDR から DOCSIS バージョンへのマッピングを含んでいる外部ファイルの名前です。

このマッピング ファイルには `giaddr-docsis-map.txt` と名前を付けて、RDU に追加する必要があります。次の方法で、`giaddr-docsis-map.txt` ファイルを RDU に追加できます。

- `Configuration.addFile()` コールを介した API。
- 管理者のユーザ インターフェイス (**Configuration > Files** を選択してアクセス)。[P.13-19](#) の「[ファイルの追加](#)」を参照してください。

*giaddr-docsis-map.txt* ファイルには、次の形式で必要な情報を含める必要があります。

```
IPv4_dotted_decimal_address_string,DOCSIS_version_string
```

- *IPv4\_dotted\_decimal\_address\_string* : CMTS インターフェイスの IP アドレスを指定します。
- *docsis\_version\_string* : ケーブル モデムがサポートする DOCSIS バージョンを示します。

サービス レベル拡張は、DHCP パケットに含まれる GIADDR アドレスを使用して、CMTS の DOCSIS バージョンを検索します。マッピング ファイルに GIADDR がない場合、拡張は */docsis/cmts/version/default* プロパティの値をケーブル モデムの DOCSIS バージョンの値に使用します。このプロパティのデフォルト値は **1.0** です。

*giaddr-docsis-map.txt* ファイルを動的に更新するには、*replaceExternalFile* API または管理者のユーザ インターフェイスを使用してファイルを編集して RDU 内で置き換えます。



(注) DOCSIS バージョンの選択のプロパティがサービス クラスで指定されていない場合、元のファイルが使用され、ネットワーク全体で体系的なアップグレードが可能になります。

### DOCSIS バージョンに基づいたサービス レベルの選択

モデムの DOCSIS バージョンと CMTS を判別後、サービス レベル選択拡張はサポートされる最小限の DOCSIS バージョンを決定して、サービス レベルに */docsis/version* プロパティを設定します。このプロパティの値は、DOCSIS バージョン文字列 (1.1 など) に設定されます。



(注) DOCSIS バージョンは、設定ファイルユーティリティを使用して指定できます。詳細については、[P.5-23 の「設定ファイルユーティリティの使用方法」](#)を参照してください。このファイルユーティリティが実行する機能は、RDU DOCSIS Version Selector 機能が CMTS でサポートされる最新の DOCSIS バージョンを判別する RDU の検証とは異なります。

### DOCSIS バージョンに基づいた DOCSIS 設定ファイル

BAC は、DOCSIS バージョンを使用して、モデムに送信する DOCSIS 設定ファイルの名前を判別します。

次のサービス クラス プロパティが、BAC の管理者のユーザ インターフェイスと API でサポートされます。

```
/cos/docsis/file/1.0
/cos/docsis/file/1.1
/cos/docsis/file/2.0
/cos/docsis/file/3.0/IPv4
/cos/docsis/file/3.0/IPv6
```

DOCSIS 設定ファイル名を特定の DOCSIS バージョンと関連付けるために、これらのプロパティをオプションで DOCSIS サービス クラスに追加できます。これらの各プロパティを設定すると、既存の DOCSIS 設定ファイル名プロパティで行われるように、RDU がサービス クラスとプロパティ値で指定されたファイルとの間にデータベース関係を確立します。

DOCSIS バージョン プロパティが存在する場合、BAC はそのプロパティ値から得られる DOCSIS バージョン文字列を、DOCSIS 設定ファイル名を提供するプロパティの名前に追加してプロパティ名を作成します。

```
/cos/docsis/file/docsis_version_string
```

サービス レベル拡張は、モデムのプロパティ階層内でこのプロパティ名を検索します。DOCSIS バージョンプロパティが見つかった場合は、プロパティ値を DOCSIS 設定ファイル名として使用します。DOCSIS バージョンプロパティが見つからない場合、BAC は、DOCSIS バージョンサフィックスのない DOCSIS 設定ファイル名を使用して、デバイス構成で指定するファイル名を提供します。

## IPv6 のサポート

この BAC リリースでは、CableLabs DOCSIS 標準の最新リビジョン、DOCSIS 3.0 をサポートしています。DOCSIS 3.0 標準には、以前の DOCSIS 標準に基づき構築された主要な新機能が導入されています。次の機能があります。

- IPv6 デバイスのプロビジョニング。次のものがあります。
  - DOCSIS 準拠のケーブル モデムと CMTS
  - コンピュータ
  - 進化する OpenCable Application Platform に基づく RNG-200 STB
  - 混在 IP モードの PacketCable Multimedia Terminal Adapter (MTA; マルチメディア ターミナルアダプタ) などの、各種 eSAFE (embedded Service/Application Functional Entities) デバイスのバリエーション

BAC は、TFTP サービスと ToD サービスに対する IPv6 サポートなど、IPv6 デバイスをプロビジョニングするために必要なサービスを提供します。BAC は IPv6 デバイスの設定ファイルも処理します。

- 拡張されたアドレス指定機能

IPv6 の主な利点は、その拡張されたアドレス指定機能です。IPv6 アドレスでは、アドレス空間が 32 ビットから 128 ビットに拡大されており、事実上、無制限のネットワークとシステムを提供できます。

- ケーブル モデムの IPv6 プロビジョニングと管理。このプロビジョニングフローには、次のものが含まれます。
  - サポートされる IP モード: BAC は、IPv4、IPv6、またはデュアルスタックモード (IPv4 および IPv6 プロビジョニングモードで構成) で DOCSIS ケーブル モデムをプロビジョニングします。さまざまなモードの詳細については、[P.6-15 の「シングルスタックとデュアルスタック」](#)を参照してください。



**(注)** ケーブル モデムは、IP のプロビジョニングモードに関係なく、IPv4 と IPv6 のトラフィックを転送できます。

- DHCPv6: DOCSIS プロビジョニングフローは、IPv6 の DHCP (DHCPv6 としても知られる) の使用を指定します。DOCSIS の DHCPv6 プロビジョニングフローの詳細については、[P.6-5 の「DOCSIS DHCPv6 ワークフロー」](#)を参照してください。

BAC における IPv6 デバイスのプロビジョニングをイネーブルにする前に、システムで IPv6 をイネーブルにする必要があります。コンピュータで IPv6 のサポートをイネーブルにするには、*root* としてログインして、次のコマンドを入力します。

```
# ifconfig intf inet6 plumb up
```

*intf* は、IPv6 をイネーブルにするインターフェイスを示します。



(注) 物理イーサネット インターフェイスに加えて、ループバック インターフェイスでも `plumb` を実行してください。次に例を示します。

```
# ifconfig bge0 inet6 plumb up
# ifconfig lo0 inet6 plumb up
```

その後、次のコマンドも実行します。

```
# /usr/lib/inet/in.ndpd
# touch /etc/hostname6.intf
```

`intf` は、IPv6 をイネーブルにするインターフェイスを示します。

次の各項では、BAC の IPv6 に関連する概念について説明します。

- [IPv6 のアドレス指定 \(P.6-14\)](#)
- [シングルスタックとデュアルスタック \(P.6-15\)](#)
- [IPv6 の DHCP オプション \(P.6-15\)](#)
- [属性とオプション \(P.6-15\)](#)

## IPv6 のアドレス指定

IPv6 アドレスの長さは 128 ビットで、コロン (:) で区切られた一連の 16 ビットの 16 進数フィールドとして表現されます。16 進数の A、B、C、D、E、および F には、大文字と小文字の区別はありません。次に例を示します。

```
2031:0000:130f:0000:0000:09c0:876a:130b
```

このアドレス指定は、次のように短縮できます。

- フィールドの先頭の 0 はオプションであり、09c0 は 9c0、0000 は 0 と記述できます。
- 0 のフィールドが連続する場合は (フィールドの数にかかわらず)、:: と表現できますが、アドレス内で 1 回しか使用できません。これは、複数回使用すると、アドレスパーサーが 0 の各ブロックのサイズを特定できないという理由からです。したがって、前述のアドレスは次のように記述できます。

```
2031:0:130f::09c0:876a:130b
```

二重コロンの短縮形を使用すると、多くのアドレスを短縮できます。たとえば、`ff01:0:0:0:0:1` は `ff01::1` になります。

リンクローカルアドレスには、リンクに限定されたスコープがあり、プレフィックス `fe80::/10` を使用します。ループバックアドレスのアドレスは、`::1` です。マルチキャストアドレスは、プレフィックス `ff00::/8` で識別されます (IPv6 にはブロードキャストアドレスはありません)。

IPv6 での IPv4 互換アドレスは、:: のプレフィックスを付けた 10 進数の 4 つの IPv4 アドレスです。たとえば、`::c0a8:1e01` として解釈される IPv4 アドレスは、`::192.168.30.1` と記述できます。

## シングル スタックとデュアル スタック

RFC 4213 は、ホストとルータ内の IPv4 と IPv6 の両方のインターネット プロトコルを全面的にサポートするための技術としてデュアル スタックを定義します。IPv4 と IPv6 の両方をサポートするネットワーク スタックは、デュアル スタックと呼ばれ、デュアル スタックを実装するホストは、デュアル スタック ホストと呼ばれます。

BAC は、次の IP モードでケーブル モデムをプロビジョニングします。

- IPv4 のみ：このモードでは、ケーブル モデムは DHCPv4 サーバに、IPv4 アドレスと関連の操作パラメータを要求します。
- IPv6 のみ：このモードでは、ケーブル モデムは DHCPv6 サーバに、IPv6 アドレスと関連の操作パラメータを要求します。モデムは IPv6 アドレスを使用して、現在の Time-of-Day と設定ファイルを取得します。
- デュアル スタック：このモードでは、ケーブル モデムは IPv6 と IPv4 の両アドレス、およびパラメータを、DHCPv6 と DHCPv4 経由でほぼ同時に取得し、IPv6 アドレスの使用優先順位を上げて、Time-of-Day と設定ファイルを取得します。



**(注)** BAC は、ケーブル モデムがプロビジョニングされている直近の IP モードで検出されたデータのみ保存します。そのため、デバイスを DHCPv4 でブートし、その後 DHCPv6 でブートすると、DHCPv6 のデータのみ検出されて保存されます。

IPv4 および IPv6 モードでプロビジョニングしている間、ケーブル モデムは、所定の時間に IP アドレス タイプ (v4 または v6) の 1 つでのみ動作します。このような理由で、プロビジョニングの IPv4 および IPv6 モードは、シングル スタック モードと呼ばれます。

デュアル スタック モードでは、IPv4 と IPv6 のアドレスを同時に使用してケーブル モデムを管理できます。このモードでは、モデムは動作可能になった後に 2 番目の IP アドレスを取得します。この機能を使用して、DOCSIS ネットワークでの IPv4 から IPv6 への能率的な移行を行えます。

## IPv6 の DHCP オプション

DOCSIS 3.0 標準は、DHCPv4 と DHCPv6 の複数の新規オプションを定義します。DHCPv6 オプションは、DHCPv4 オプションは使用しません。DHCPv6 のオプションは固有の別オプションです。BAC がサポートする DHCPv6 オプションのリストについては、『*User Guide for Cisco Network Registrar 7.0*』を参照してください。

## 属性とオプション

この項では、BAC 4.0 が Network Registrar と通信するとき使用する属性とオプションについて説明します。

BAC は、Network Registrar にインストールされている DHCP 拡張を使用して、そのデータベース内の設定に基づき DHCP メッセージを操作します。これらの拡張を使用して、BAC は DHCP 要求から情報を取得し、DHCP 応答に値を設定します。このようにして、BAC はプロビジョニングするデバイスにカスタマイズされた設定を提供します。

このインタラクションを容易にするために、Network Registrar は、辞書のセットを BAC 拡張に公開します。BAC 拡張は、これらの辞書を使用して Network Registrar と対話します。

辞書には、要求辞書と応答辞書が使用する属性辞書、環境辞書、および通知辞書の 3 種類があります。

- 環境辞書：DHCP サーバが拡張と通信するために使用する辞書に含まれる属性を表します。
- 要求辞書：要求パケットの DHCP オプションと属性を表します。
- 応答辞書：応答パケットの DHCP オプションと属性を表します。
- 通知辞書：BAC 拡張と RDU 間で伝達される情報を表します。

辞書は、BAC と Network Registrar で設定されているさまざまな DHCP オプションと設定を表します。オプションは、DHCP メッセージのオプション フィールドに保存されている DHCP 設定パラメータと他の制御情報です。DHCP クライアントは、要求されているオプションを判別して、DHCP パケットで送信します。

属性は名前と値のペアで、次のようなものがあります。

- DHCPv4 オプション。たとえば、**relay-agent-info**。
- DHCPv4 オプションから取得する情報のサブセット。たとえば、**relay-agent-remote-id** は、DHCPv4 Option 82 のサブオプション 2 を表します。
- DHCPv4 オプションのフィールド。たとえば、「file」は DHCPv4 のヘッダーフィールドです。

属性には次のような設定も含めることもできます。

- Network Registrar の動作を制御する設定。たとえば、「drop」はパケットが破棄されることを示します。
- 情報を提供する設定。

BAC 4.0 と Network Registrar 7.0 では、次の 2 つの API バージョンをサポートします。BAC 拡張はそれらの API を使用して、DHCPv4 または DHCPv6 をイネーブルにします。

- DEX API バージョン 1: この API を使用すると、Network Registrar 拡張は、属性を介して DHCPv4 パケットの詳細をクエリーできます。
- DEX API バージョン 2: この API を使用すると、Network Registrar 拡張は、DHCPv4 オプションと DHCPv6 オプション、およびサブオプションに直接クエリーできます。

BAC 拡張は、Network Registrar 拡張の API バージョンが DEX API バージョン 2 であることを検出すると、DHCPv6 のサポートをイネーブルにします。

### DHCPv6 用に検出されたデータを制御するプロパティ

BAC 拡張が DHCPv6 用に検出するデータを制御するプロパティには 3 種類のセットがあります。



(注) 管理者のユーザ インターフェイスを使用すると、**Configuration > Defaults > NR Defaults** ページでこれらのプロパティの設定を確認できます。

- バージョン 4.0 以前の Network Registrar 拡張の動作を制御するプロパティ。表 6-3 を参照してください。
- BAC 4.0 での DHCPv4 の Network Registrar 拡張の動作を制御するプロパティ。表 6-4 を参照してください。
- クライアント（ケーブル モデム）とリレー エージェント（CMTS）に対する、BAC 4.0 での DHCPv6 の Network Registrar 拡張の動作を制御するプロパティ。この違いは、DHCPv4 標準はクライアント メッセージとリレー メッセージを組み合わせる 1 つのメッセージにし、それに対して、DHCPv6 標準はそれらのメッセージを分割するという点にあります。表 6-5 を参照してください。

表 6-3 は、BAC のバージョン 4.0 以前での Network Registrar 拡張の動作に影響を与えるプロパティについて説明しています。



表 6-3 BAC 4.0 以前の Network Registrar 拡張のプロパティ

プロパティ名	説明
<i>/cnrExtension/attributesRequiredInRequest</i>	RDU に構成生成の要求を送信するために、Network Registrar 要求辞書が拡張に含める必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_REQUIRED_IN_REQUEST_DICTIONARY
<i>/cnrExtension/attributesToPullFromRequestAsBytes</i>	Network Request 要求辞書からバイナリ形式でプルする必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_REQUEST_DICTIONARY_AS_BYTES
<i>/cnrExtension/attributesToPullFromRequestAsStrings</i>	Network Registrar 要求辞書から文字列形式でプルする必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_REQUEST_DICTIONARY_AS_STRINGS
<i>/cnrExtension/attributesToReadFromEnvironmentDictionary</i>	Network Registrar 環境辞書からプルする必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_ENVIRONMENT_DICTIONARY

表 6-4 では、BAC 4.0 での DHCPv4 の Network Registrar 拡張の動作を制御するプロパティについて説明しています。

表 6-4 BAC 4.0 の DHCPv4 Network Registrar 拡張のプロパティ

プロパティ名	説明
<i>/cnrExtension/attributesRequiredInV4Request</i>	RDU に構成生成の要求を送信するために、Network Registrar 要求辞書が拡張に含める必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_REQUIRED_IN_V4_REQUEST_DICTIONARY
<i>/cnrExtension/attributesToPullFromV4RequestAsBytes</i>	Network Registrar 要求辞書からバイナリ形式でプルする属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_BYTES
<i>/cnrExtension/attributesToPullFromV4RequestAsStrings</i>	Network Registrar 要求辞書から文字列形式でプルする属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_STRINGS

表 6-5 では、BAC 4.0 での DHCPv6 の Network Registrar 拡張の動作を制御するプロパティについて説明しています。

表 6-5 BAC 4.0 の DHCPv6 Network Registrar 拡張のプロパティ

プロパティ名	説明
<b>クライアント メッセージ</b>	
<i>/cnrExtension/attributesRequiredInV6 Request</i>	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 要求辞書が拡張に含める必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_REQUIRED_IN_V6_REQUEST_DICTIONARY
<i>/cnrExtension/attributesToPullFromV6 RequestAsBytes</i>	Network Registrar DHCPv6 要求辞書からバイナリ形式でプルする属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES
<i>/cnrExtension/optionsRequiredInV6 Request</i>	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 要求辞書が拡張に含める必要がある DHCP オプションのリストを示します。 <b>API 定数</b> CNR_OPTIONS_REQUIRED_IN_V6_REQUEST_DICTIONARY
<i>/cnrExtension/optionsToPullFromV6 Request AsBytes</i>	Network Registrar DHCPv6 要求辞書からバイナリ形式でプルする DHCP オプションのリストを示します。 <b>API 定数</b> CNR_OPTIONS_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES
<b>リレー メッセージ</b>	
<i>/cnrExtension/attributesRequiredInV6 Relay</i>	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 Relay-Forward 要求辞書が拡張に含める必要がある属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_REQUIRED_IN_V6_RELAY_DICTIONARY
<i>/cnrExtension/attributesToPullFromV6 RelayAsBytes</i>	Network Registrar DHCPv6 Relay-Forward 要求リレー辞書からバイナリ形式でプルする属性のリストを示します。 <b>API 定数</b> CNR_ATTRIBUTES_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES
<i>/cnrExtension/optionsRequiredInV6Relay</i>	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 Relay-Forward 要求辞書が拡張に含める必要がある DHCP オプションのリストを示します。 <b>API 定数</b> CNR_OPTIONS_REQUIRED_IN_V6_RELAY_DICTIONARY
<i>/cnrExtension/optionsToPullFromV6 RelayAsBytes</i>	Network Registrar DHCPv6 Relay-Forward 要求リレー辞書からバイナリ形式でプルする DHCP オプションのリストを示します。 <b>API 定数</b> CNR_OPTIONS_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES

## IPv6 の設定ワークフロー

IPv6 をサポートするための BAC の設定には、2 つの異なるワークフローがあります。

- プロビジョニング グループでの DPE の設定。表 3-3 を参照してください。
- ネットワークでの Network Registrar サーバの設定。表 3-5 を参照してください。

## リース クエリー

BAC RDU は、DHCP リース クエリー プロトコルを使用して、Network Registrar に対してデバイスの IP アドレスをクエリーします。その後、BAC はこの情報を IPv4 と IPv6 の両デバイスのデバイス中断や詳細なレポート作成に使用します。

この BAC リリースは、次の構成をサポートしています。

- リース クエリーの自動設定 (P.6-19)
- リース クエリーの送信元 IP アドレス (P.6-19)

## リース クエリーの自動設定

RDU は名前解決を実行して、リース クエリーの送信先である Network Registrar サーバの IP アドレスを決定します。DNS で障害が発生すると、リース クエリーは失敗します。この BAC リリースでは、RDU がリース クエリー要求を送信する必要がある、プロビジョニング グループ内の Network Registrar サーバの IP アドレスを直接設定できます。

自動設定をイネーブルにすると、RDU はそのリース クエリーの設定を調整して、プロビジョニング グループ内の Network Registrar サーバから IPv4 と IPv6 の両方のアドレス リストを設定します。RDU は、サーバに現在登録されている情報と RDU データベースに格納されている情報を比較してから、このタスクを実行します。BAC Network Registrar 拡張が 1 つのプロビジョニング グループから別のグループに移動している場合、リース クエリーの設定は変更され、次の内容が削除されます。

- 以前のプロビジョニング グループ オブジェクトに対するリース クエリーの設定に存在する IP アドレス。
- IP アドレス リストにもう存在しない IP アドレス。

RDU は、リース クエリーの設定を検索して、プロビジョニング グループが指定した拡張を使用するように設定されているかどうかを検証します。プロビジョニング グループが拡張を使用するように設定されていない場合、RDU は Network Registrar サーバに登録されているアドレスからアドレスを選択して、プロビジョニング グループのリース クエリーの設定に追加します。

この自動設定をディセーブルにした場合、RDU は Network Registrar サーバでの登録時にそのリース クエリーの設定を変更しません。デフォルトでは、この機能はイネーブルになっています。

プロビジョニング グループ内のリース クエリーアドレスの自動設定をイネーブルまたはディセーブルにするには、管理者ユーザ インターフェイスから LeaseQuery AutoConfig オプションを設定できます。P.12-29 の「プロビジョニング グループの表示」を参照してください。

## リース クエリーの送信元 IP アドレス

以前のバージョンの BAC では、リース クエリー機能は、リース クエリー要求を送信するための送信元インターフェイスと送信元ポートの選択をオペレーティング システムに依存していました。このリリースではこれがデフォルトの動作ですが、特定のインターフェイスを使用してリース クエリー要求を送信するように RDU を設定することもできます。

## リース クエリーの設定

デフォルトでは、BAC は表 6-1 に記載されている IP アドレスとポートにバインドします。

表 6-1 バインディング用のリース クエリー アドレス

プロトコル	IP アドレス	ポート
IPv4	ワイルドカード <sup>1</sup>	67
IPv6	ワイルドカード	547

1. ワイルドカードは、特別なローカル IP アドレスです。通常は「すべて」を意味し、バインド操作でのみ使用できます。

RDU でポート 547 とポート 67 が利用可能な場合は、リース クエリー要求を送信するために特別な設定を行う必要はありません。RDU のインストール中に、インストールプログラムがこれらのポートのいずれかが他のプロセスで使用されていることを検出した場合、オペレーティング システムが選択する動的ポートを使用することをお勧めします。

次に例を示します。

```
DHCPv4/DHCPv6 lease query port(s) (Udp/67 and Udp/547) is in use.
Configuring the RDU to use a dynamic port for DHCPv4/DHCPv6 lease query.
```

インストール プログラムは、*BPR\_HOME/rdu/conf/rdu.properties* ファイルの次のプロパティに値 0 を設定して動的ポートの選択を自動的にイネーブルにします。

```
/cnrQuery/clientSocketAddress=0.0.0.0:0
/cnrQuery/ipv6/clientSocketAddress=[::]:0
```

同じプロパティを使用して、リース クエリーの通信に使用する IP アドレスとポートを設定することもできます。次に例を示します。

```
/cnrQuery/clientSocketAddress=10.1.2.3:166
/cnrQuery/ipv6/clientSocketAddress=[2001:0DB8:0:0:203:baff:fe12:d5ea]:1547
```

RDU は、これらのプロパティを使用して、ユーザが指定する IP アドレスとポートにバインドします。



(注) *rdu.properties* ファイルのプロパティを手動で変更する場合は、必ず RDU を再起動してください。  
`/etc/init.d/bprAgent restart rdu` コマンドを使用します。

## リース クエリーのリレー エージェントとしての BAC の設定

リレー エージェントとして動作するように BAC を設定できます。リレー エージェントのオプションは次のとおりです。

- IPv4 ではデフォルトでイネーブル化
- IPv6 ではデフォルトでディセーブル化

### IPv4 のリース クエリーの場合

BAC が IPv4 リース クエリーのリレー エージェントとして動作する場合、BAC はリース クエリー要求パケットに GIADDR (DHCP サーバが応答する IP アドレス) を提供します。デフォルトでは、RDU は、この目的でコンピュータのプライマリ IP アドレスを使用します。



(注) 配備内のすべての DHCP サーバがこの IP アドレスにアクセスできるようにしてください。また、このプロパティで使用する IP アドレスは、RDU をインストールしたコンピュータに存在する必要があります。

GIADDR フィールドで使用する IP アドレスを変更するには、*rdu.properties* ファイルの */cnrQuery/giaddr* プロパティの値を変更する必要があります。たとえば、GIADDR を 10.10.10.1 に変更する場合、次のように追加します。

```
/cnrQuery/giaddr=10.10.10.1
```

*rdu.properties* ファイルのプロパティを手動で変更する場合は、*/etc/init.d/bprAgent restart rdu* コマンドを使用して RDU を再起動してください。

## IPv6 リース クエリーの場合

BAC が IPv6 リース クエリーのリレー エージェントとして動作するように設定するには、*rdu.properties* ファイルに次のプロパティを含める必要があります。

```
/cnrQuery/ipv6/linkAddress=IPv6 address
```

```
/cnrQuery/ipv6/peerAddress=IPv6 address
```

次に例を示します。

```
/cnrQuery/ipv6/linkAddress=2001:0DB8:0:0:203:baff:fe12:d5ea
/cnrQuery/ipv6/peerAddress=2001:0DB8:0:0:203:baff:fe12:d5ea
```



(注) リンク アドレスとピア アドレスに入力する値は、BAC と Network Registrar を稼動するネットワークの構成によって異なります。単純なケースでは、リンク アドレスとピア アドレスを RDU ホストの IPv6 アドレスに設定する必要があります。この IPv6 アドレスは Network Registrar にルーティング可能である必要があります。

*/etc/init.d/bprAgent restart rdu* コマンドを使用して、RDU を再起動します。

## 例

この例は、リレー エージェント オプションがイネーブルになっている IPv6 リース クエリー要求の出力を示しています。

```
rdu.example.com: 2007 10 18 19:40:30 EDT: %BAC-RDU-7-DEBUG_DHCP_IF_IPV6:
PACE-2:ServerBatch[Batch:rdu.example.com/10.10.10.1:1b994de:115b52abeb4:80000278]:
Peer[rdu.example.com:33743]: Querying single prov group for DUID
[00:03:00:01:23:45:67:89:98:56] via DHCPv6 LEASEQUERY packet [version V6, message-type
12, hop-count 0, link-address 2001:0DB8:0:0:203:baff:fe12:d5ea, peer-address
2001:0DB8:0:0:203:baff:fe12:d5ea, (relay_msg (9) option (52 bytes) version V6,
message-type 14, transaction-id 13401290, (client-identifier (1) option (9 bytes)
00:11:22:33:44:55:66:77:88), (lq-query (44) option (31 bytes) query-type 2,
link-address 0:0:0:0:0:0:0:0, (client-identifier (1) option (10 bytes)
00:03:00:01:23:45:67:89:98:56)))]
```

## AIC Echo のイネーブル化

AIC Echo オプションを使用して、標準ポートではなく、要求元のクライアントの送信元ポートに回答を送信するように Network Registrar を設定できます。

たとえば、IP アドレスが 10.1.1.1 のクライアントがポート 1456 を使用して要求を転送し、サーバ上で AIC Echo がディセーブルになっている場合、そのサーバは回答を標準クライアントポートに返します。プロトコルスタックに応じて、標準クライアントポートは次のいずれかになります。

- 67 は IPv4
- 546 は IPv6

AIC Echo がイネーブルになっている場合、回答はポート 1456 に転送されます。

IPv4 リース クエリーを要求する場合、AIC Echo はデフォルトでディセーブルになります。このオプションは、デフォルトの IPv4 バインディングポートが変更された場合にのみ使用されます。

IPv6 リース クエリーを要求する場合、AIC Echo はデフォルトでイネーブルになります。ただし、IPv6 リース クエリー メッセージはデフォルトではリレーされないため、このオプションは、標準クライアントポートの 546 ではなく、ポート 547 にリース クエリーの回答を返すために使用されます。

## リース クエリーのデバッグ

RDU で情報レベル ログイング (6 : 情報) を使用して、リース クエリー処理に関連する重要な詳細情報を表示できます (RDU でのログ レベルを設定するには、[P.10-5](#) の「[RDU ログ レベル ツールの使用方法](#)」を参照してください)。

リース クエリー機能をデバッグする場合は、次のプロパティを使用できます。

- `dhcpleasequeryv4` : IPv4 リース クエリーのデバッグ
- `dhcpleasequeryv6` : IPv6 リース クエリーのデバッグ

## IPv6 リース クエリーの使用例

この項では、次の IPv6 リース クエリーの使用例について説明します。

- プロビジョニング グループ内の全 (2 台) Network Registrar サーバにおけるクライアントごとに 1 つのリース
- プロビジョニング グループ内の全 (2 台) Network Registrar サーバにおけるクライアントごとに複数のリース
- 1 台の Network Registrar サーバのクライアントごとに複数のリース
- 委任プレフィックスがあるデバイスのリース

### プロビジョニング グループ内の全 (2 台) Network Registrar サーバにおけるクライアントごとに 1 つのリース

IPv6 に対するフェールオーバー プロトコルがまだ定義されていない状態では、通常、プロビジョニング グループ内の 1 台の Network Registrar サーバのみがクライアントのリース情報を保持します。このケースで、プロビジョニング グループに 2 台の Network Registrar サーバが存在する場合、RDU は両方のサーバにリース クエリー要求を送信しますが、1 台からのみ回答を受信します。IP アドレスはその回答で指定されたものが使用されます。

この IP アドレスは、次の方法で表示できます。

- 管理者のユーザ インターフェイス (**Devices > Device Details** ページ上)。
- API (リース クエリー マップで `client-ipaddress` 属性を使用)。

## プロビジョニング グループ内の全 (2 台) Network Registrar サーバにおけるクライアントごとに複数のリース

まれに、プロビジョニング グループ内の両方の Network Registrar サーバが同じクライアントのリースを保持し、両方のサーバがリース クエリー応答に回答することがあります。この場合、DHCPv6 Leasequery ドラフトごとに、直近の OPTION\_CLT\_TIME (client-last-transaction-time) がある応答が使用されます。

## 1 台の Network Registrar サーバのクライアントごとに複数のリース

クライアントが同じサーバの異なる 2 つのリンクにリースを保持している場合、Network Registrar では、応答時の OPTION\_LQ\_CLIENT\_LINK オプションにすべてのリンク アドレスが含まれます。その後、BAC は Network Registrar に個々のリンクをクエリーして、すべての IP アドレスを取得します。このリストでは、BAC は、ループバックでもデバイス中断用のマルチキャスト アドレスでもない最初の IP アドレスを使用します。

管理者のユーザ インターフェイスを使用して、このプロセスで取得した IP アドレスのリストを **Devices > Device Details** ページで表示できます。

## 委任プレフィックスがあるデバイスのリース

IP アドレス、または委任プレフィックス、あるいは両方が割り当てられているデバイスのリース クエリー要求を送信できます。

管理者のユーザ インターフェイスを使用して、IP アドレスとプレフィックスを **Devices > Device Details** ページで表示できます。この IP アドレスを API を使用して取得するには、リース クエリー マップで `iaprefix` 属性を使用します。

