Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ ブロードバンド ネットワーク ゲートウェイ コンフィギュレーション ガイド リリース 4.3.x

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

Text Part Number: OL-28375-01-J

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

はじめに

この章で説明する内容は、次のとおりです。

マニュアルの変更履歴

この表に、初版後、このマニュアルに加えられた技術的な変更の履歴を示します。

リビジョン	日付	概要
OL-28375-01-J	2012 年 12 月	このマニュアルの初版

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

第 1 章

Cisco IOS XR リリース 4.3.x の新機能および変更された機能情報

次の表では、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*』における新機能および変更された機能に関する情報を要約し、その参照先を示しています。

Cisco IOS XR Software, Release 4.3.x の新機能および変更された機能の完全なリストについては、『New and Changed Features in Cisco IOS XR Software, Release 4.3.x for Cisco ASR 9000 Series Aggregation Services Router』マニュアルを参照してください。

新機能および変更された機能に関する情報

次の表では、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*』における新機能および変更された機能に関する情報を要約し、その参照先を示しています。

表 *1*：新機能および変更された機能

機能	説明	導入/更新されたリリース	参照先
PPPoE スマート サーバ選択	この機能が導入されました。	リリース 4.3.1	「加入者セッションの確立」の章： • PPPoE スマート サーバ選択, （138 ページ） • PADO 遅延の設定, （139 ページ） 「*BNG* 機能の *XML* サポート」の章： • PPPoE XML サポート, （360 ページ） PPPoE スマート サーバ選択機能の PADO 遅延の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*PPPoE Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
インターフェイスまたは VLAN サブインターフェイスの NAS-Port-Type	NAS-Port-Type は、インターフェイスまたは VLAN サブインターフェイスで設定可能になりました。	リリース 4.3.1	「認証、許可、アカウンティング機能の設定」の章： • インターフェイスまたは VLAN サブインターフェイスの NAS-Port-Type, （36 ページ） • RADIUS 属性の NAS-Port-Type の設定, （44 ページ） NAS-Port-Type の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*BNG AAA Commands*」の章を参照してください。
LAC での L2TP の再構築	この機能が導入されました。	リリース 4.3.1	「加入者セッションの確立」の章： • LAC での L2TP の再構築, （117 ページ） • LAC での L2TP の再構築のイネーブル化, （118 ページ） NAS-Port-Type の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*PPPoE LAC-Specific Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
LAC の IETF タグ付き属性	タグのサポートは、LAC の IETF 属性に追加されました。	リリース 4.3.1	付録 B「*RADIUS* 属性」の章： • LAC の IETF タグ付き属性, （369 ページ）
PPPoE セッション制限およびスロットル	PPPoE セッションの制限とスロットルのサポートは、制限のしきい値と、スロットルの要求数、要求期間、およびブロック期間のしきい値を追加することによって、より多くのパラメータに拡張されました。	リリース 4.3.1	「加入者セッションの確立」の章： • PPPoE セッション制限, （141 ページ） • PPPoE セッション スロットル, （144 ページ） • PPPoE セッション制限の設定, （142 ページ） • PPPoE セッション スロットルの設定, （144 ページ） PPPoE セッションの制限とスロットルの設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*PPPoE Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
あいまいな VLAN	あいまいな VLAN の設定は、2 つの新しいカプセル化をサポートするように拡張されました。	リリース 4.3.1	「加入者セッションの確立」の章： • あいまいな VLAN の設定, （203 ページ） 「加入者機能の設定」の章： • あいまいな VLAN での加入者セッションの確立, （293 ページ）
ASR 9922 のサポート	BNG のサポートは、Cisco ASR 9922 シリーズ アグリゲーション サービス ルータに追加されます。	リリース 4.3.1	「ブロードバンド ネットワーク ゲートウェイの概要」の章： • BNG のハードウェア要件, （22 ページ）
DHCPv6 システム リロードの持続的なバインディング	この機能が導入されました。	リリース 4.3.1	「加入者セッションの確立」の章： • DHCPv6 システム リロードの持続的なバインディング, （181 ページ） DHCP コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*BNG DHCP Commands*」の章を参照してください。
Circuit-ID 単位のセッション制限	AAA からの Circuit-ID 単位のセッション制限を受け入れ、プロファイルから設定済みのリース制限を上書きするためのサポートが追加されました。	リリース 4.3.1	「加入者セッションの確立」の章： • DHCP リース制限の指定, （163 ページ）

機能	説明	導入/更新されたリリース	参照先
アクセス インターフェイスの VLAN ポリシー	この機能が導入されました。	リリース 4.3.1	「*Quality of Service*（*QoS*）の導入」の章： • アクセス インターフェイスの VLAN ポリシー, （261 ページ） • アクセス インターフェイスでの VLAN ポリシーの設定, （264 ページ） DHCP コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*QOS Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
DHCPv6	この機能は、IPv6 でサポートされました。	リリース 4.3.0	「加入者セッションの確立」の章： • DHCPv6 の概要, （170 ページ） • DHCPv6 サーバおよび DHCPv6 リレーまたはプロキシ, （171 ページ） • DHCPv6 機能, （180 ページ） • 異なるコンフィギュレーション モードの DHCPv6 のイネーブル化, （172 ページ） • DHCPv6 パラメータの設定, （178 ページ） • IPv6 IPoE 加入者インターフェイスの設定, （182 ページ） • IPv6 PPPoE 加入者インターフェイスの設定, （193 ページ） • IPv6 アドレスまたはプレフィックス プール名の設定, （206 ページ） DHCP IPv6 の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*BNG DHCP Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
DS-Lite	この機能が導入されました。	リリース 4.3.0	「加入者セッションの確立」の章： • DHCPv6 Dual-Stack Lite サポート, （211 ページ） • DS-Lite の AFTR 完全修飾ドメイン名の設定, （212 ページ）
IPv6 ネイバー探索	この機能は、IPv6 でサポートされました。	リリース 4.3.0	「加入者セッションの確立」の章： • IPv6 ネイバー探索, （217 ページ） IPv6 ND の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*Neighbor Discovery Commands*」の章を参照してください。
BNG パッケージ	この機能が導入されました。	リリース 4.3.0	「ブロードバンド ネットワーク ゲートウェイの概要」の章： • BNG パッケージ, （20 ページ） • Cisco ASR 9000 ルータでの BNG PIE の構築およびインストール, （20 ページ）

機能	説明	導入/更新されたリリース	参照先
HTTP-Redirect での IPv6 サポート	この機能は、IPv6 でサポートされました。	リリース 4.3.0	「加入者機能の設定」の章： • PBR を使用した HTTP リダイレクト, （328 ページ） • リダイレクションに対する HTTP の宛先の識別, （330 ページ） • HTTP リダイレクションのクラス マップの設定, （335 ページ） • HTTP リダイレクトのポリシー マップの設定, （337 ページ） • HTTPR ポリシーを適用するための動的なテンプレートの設定, （340 ページ）
IPv6 over PPPoE/IPoE セッションおよび IPoE/PTA の IPv6 uRPF	この機能が導入されました。	リリース 4.3.0	「加入者セッションの確立」の章： • IPoE セッションの確立, （93 ページ） • PPPoE セッションの確立, （105 ページ） • uRPF, （296 ページ） IPv6 over IPoE/PPPoE の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*IPoE Commands*」の章と「*PPPoE Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
BNG 衛星拡張	この機能が導入されました。	リリース 4.3.0	「ブロードバンド ネットワーク ゲートウェイの概要」の章： • BNG の相互運用性機能
DAPS での IPv6 サポート	この機能は、IPv6 でサポートされました。	リリース 4.3.0	「加入者機能の設定」の章： • DAPS サポート, （313 ページ） • アドレス プールのサブネット番号およびマスクの設定, （318 ページ） • IPv6 アドレスの範囲の指定, （320 ページ） • 使用率のしきい値の指定, （322 ページ） • プレフィックスの長さの指定, （324 ページ） • サブネット内の一連のアドレスまたはプレフィックスの指定, （326 ページ） DAPS の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*Address Pool Service Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
加入者インターフェイス拡張のパケット処理	この機能は、拡張によってサポートされました。	リリース 4.3.0	「加入者セッションの確立」の章： • 加入者インターフェイスでのパケット処理, （215 ページ）
QoS：共有ポリシーインスタンスのサポート	この機能が導入されました。	リリース 4.3.0	「*QoS* の導入」の章： • 共有ポリシー インスタンスのサポート, （242 ページ） • サービスポリシーの設定および動的なテンプレートを使用した加入者設定の適用, （225 ページ） • RADIUS を使用した入力または出力方向での SPI を持つポリシーの設定, （248 ページ） QoS の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*QoS Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
ACL での IPv6 サポート	この機能は、IPv6 でサポートされました。	リリース 4.3.0	「加入者機能の設定」の章： • アクセス コントロール リストおよびアクセス コントロール リストベース転送, （274 ページ） • アクセスコントロール リストの設定, （275 ページ） • ACL のアクティブ化, （277 ページ） QoS の設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*ACL and ABF Commands*」の章を参照してください。

機能	説明	導入/更新されたリリース	参照先
過剰なパント フロー トラップ	この機能が導入されました。	リリース 4.3.0	「加入者機能の設定」の章： • 過剰なパント フロー トラップ, （269 ページ） • 過剰なパント フロー トラップ処理のイネーブル化, （272 ページ） 過剰なパント フロー トラップの設定に使用するコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』の「*Excessive Punt Flow Trap Commands*」の章を参照してください。
TCP MSS の処理	この機能が導入されました。	リリース 4.3.0	「加入者機能の設定」の章： • TCP MSS 調整, （288 ページ） • TCP パケットの TCP MSS 値の設定, （290 ページ）
RADIUS ベースの合法的傍受	この機能が導入されました。	リリース 4.3.0	「加入者機能の設定」の章： • RADIUS ベースの合法的傍受, （285 ページ） • RADIUS ベースの合法的傍受のイネーブル化, （286 ページ）

第 2 章

ブロードバンド ネットワーク ゲートウェイの概要

この章では、Cisco ASR 9000 シリーズ ルータに実装されているブロードバンド ネットワーク ゲートウェイ（BNG）機能の概要を説明します。

BNG について

ブロードバンド ネットワーク ゲートウェイ（BNG）は、ブロードバンド ネットワークに接続する加入者用のアクセス ポイントです。 接続が BNG と宅内装置（CPE）間で確立されている場合、加入者は、ネットワーク サービス プロバイダー（NSP）またはインターネット サービス プロバイダー（ISP）が提供するブロードバンド サービスにアクセスできます。

BNG は、加入者セッションを確立および管理します。 セッションがアクティブな場合、BNG はアクセス ネットワークのさまざまな加入者セッションからのトラフィックを集約し、サービス プロバイダーのネットワークにルーティングします。

BNG は、サービス プロバイダーによって導入され、エッジ ルータなどのネットワークの最初の集約ポイントに存在します。 Cisco ASR 9000 シリーズ ルータなどのエッジ ルータは、BNG として機能するように設定する必要があります。 加入者はエッジ ルータに直接接続しているため、BNG は加入者アクセスを効果的に管理します。加入者管理機能は、次のとおりです。

- 加入者セッションの認証、許可、アカウンティング
- アドレス割り当て
- セキュリティ
- ポリシー管理
- Quality of Service（QoS）

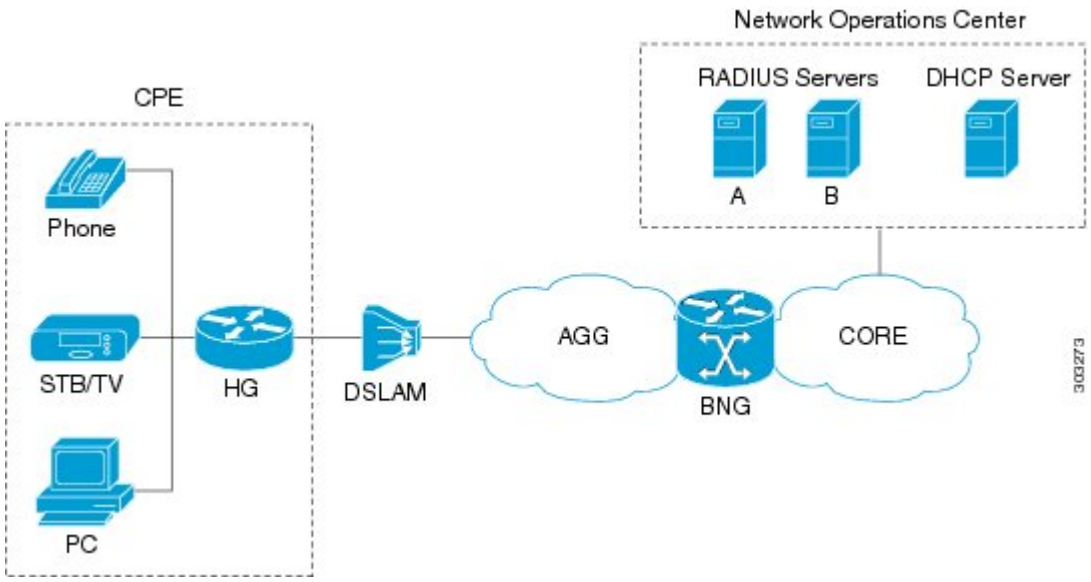
BNG を使用するメリットは、次のとおりです。

- BNG ルータは、ルーティング機能を実行するだけでなく、認証、許可、アカウンティング（AAA）サーバと通信してセッション管理機能と課金機能も実行します。 これにより、BNG ソリューションはさらに包括的になります。
- 異なる加入者を異なるネットワーク サービスに提供できます。 これにより、サービス プロバイダーは、必要に応じて各カスタマーのブロードバンドパッケージをカスタマイズできます。

BNG アーキテクチャ

BNGアーキテクチャの目的は、ブロードバンド接続を加入者に提供し、加入者セッションを管理するために、BNG ルータが周辺機器（CPE など）およびサーバ（AAA および DHCP など）と対話できるようにすることです。 次の図は、基本的な BNG アーキテクチャを示しています。

図 *1*：*BNG* アーキテクチャ



BNG アーキテクチャは、次のタスクを実行するように設計されています。

- ブロードバンド サービスを供給される必要がある宅内装置（CPE）との接続。

- IPoE または PPPoE プロトコルを使用した加入者セッションの確立。
- 加入者を認証し、加入者セッションのアカウントを保持する AAA サーバとの対話。
- クライアントに IP アドレスを提供する DHCP サーバとの対話。

4 つの BNG タスクは、次の項で簡単に説明されています。

CPE との接続

BNG は、マルチプレクサおよびホーム ゲートウェイ（HG）経由で CPE に接続します。 CPE は、通信、つまり音声（電話）、ビデオ（セット トップ ボックス）、およびデータ（PC）でトリプル プレイ サービスを表します。 個々の加入者デバイスは、HG に接続します。 この例では、加入者はデジタル加入者線（DSL）接続を介してネットワークに接続します。 したがって、HG は DSL アクセス マルチプレクサ（DSLAM）に接続します。

複数の HG は、BNG ルータに集約されたトラフィックを送信する単一の DSLAM に接続できます。 BNG ルータは、ブロードバンド リモート アクセス デバイス（DSLAM またはイーサネット アグリゲーション スイッチなど）とサービス プロバイダー ネットワーク間のトラフィックをルーティングします。

加入者セッションの確立

各加入者（または、具体的には CPE で実行されているアプリケーション）は、論理セッションによってネットワークに接続します。 使用されるプロトコルに基づいて、加入者セッションは 2 つに分類されます。

- PPPoE 加入者セッション：PPP over Ethernet（PPPoE）加入者セッションは、CPE と BNG 間で実行されるポイントツーポイント（PPP）プロトコルを使用して確立されます。
- IPoE 加入者セッション：IP over Ethernet（IPoE）加入者セッションは、CPE と BNG 間で実行される IP プロトコルを使用して確立されます。IP アドレッシングは、DHCP プロトコルを使用して実行されます。

RADIUS サーバとの対話

BNG は、外部のリモート認証ダイヤルイン ユーザ サービス（RADIUS）サーバに依存して、加入者に認証、許可、アカウンティング（AAA）機能を提供します。 AAA プロセス中、BNG は RADIUS を使用して次の処理を実行します。

- 加入者セッションを確立する前に、加入者を認証します。
- 特定のネットワーク サービスまたはリソースへのアクセスを加入者に許可します。
- アカウンティングまたは課金に対するブロードバンド サービスの使用を追跡します。

RADIUS サーバには、サービス プロバイダーの加入者全員の完全なデータベースが含まれており、RADIUS メッセージ内の属性の形で BNG に加入者データの更新を提供します。 一方、BNG は、RADIUS サーバにセッション使用状況（アカウンティング）の情報を提供します。 RADIUS 属性の詳細については、RADIUS 属性, （367 ページ）を参照してください。

BNG は、AAA プロセスでフェールオーバー冗長性を持つために、複数の RADIUS サーバとの接続をサポートします。 たとえばRADIUS サーバ A がアクティブの場合、BNG はすべてのメッセージを RADIUS サーバ A に渡します。 RADIUS サーバ A との通信が失われた場合、BNG はすべてのメッセージの宛先を RADIUS サーバ B に変更します。

BNG と RADIUS サーバ間の対話中、BNG はラウンドロビン方式でロード バランシングを実行します。 ロード バランシング プロセス中、RADIUS サーバ A に処理するための帯域幅がある場合にのみ、BNG は AAA 処理要求を RADIUS サーバ A に送信します。 それ以外の場合、要求は RADIUS サーバ B に送信されます。

DHCP サーバとの対話

BNG は、アドレス割り当ておよびクライアント設定機能について、外部のダイナミック ホスト コンフィギュレーション プロトコル（DHCP）サーバに依存します。 BNG は、アドレッシング プロセスでフェールオーバー冗長性を持つために、複数の DHCP サーバに接続できます。 DHCP サーバには、CPE にアドレスを割り当てる IP アドレス プールが含まれています。

BNG と DHCP サーバ間の対話中、BNG は DHCP リレーまたは DHCP プロキシとして機能します。

DHCP リレーとして、BNG はクライアント CPE から DHCP ブロードキャストを受信し、DHCP サーバに要求を転送します。

DHCP プロキシとして、BNG 自体は DHCP サーバからアドレス プールを取得することでアドレス プールを維持し、IP アドレスのリースも管理します。 BNG は、レイヤ 2 でクライアント ホーム ゲートウェイと通信し、レイヤ 3 で DHCP サーバと通信します。

DSLAM は、加入者 ID 情報を挿入することによって DHCP パケットを変更します。 BNG は、DSLAM によって挿入された ID 情報と DHCP サーバによって割り当てられたアドレスを使用し、ネットワーク上の加入者を識別して IP アドレスのリースをモニタします。

ISP ネットワーク モデルでの BNG の役割

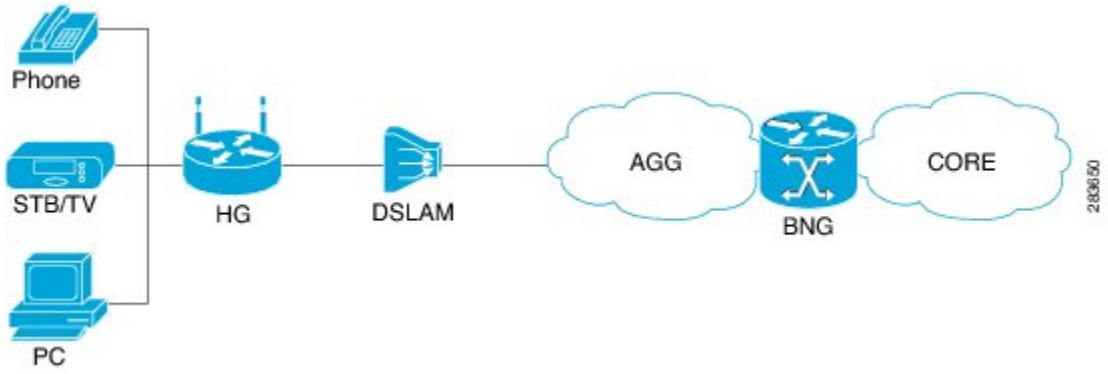
BNG の役割は、加入者から ISP にトラフィックを渡すことです。 BNG が ISP に接続する方法は、BNG が存在するネットワークのモデルによって異なります。 ネットワーク モデルには、次の 2 つのタイプがあります。

- ネットワーク サービス プロバイダー, （19 ページ）
- アクセス ネットワーク プロバイダー, （19 ページ）

ネットワーク サービス プロバイダー

次の図は、ネットワーク サービス プロバイダー モデルのトポロジを示しています。

図 *2*：ネットワーク サービス プロバイダー モデル

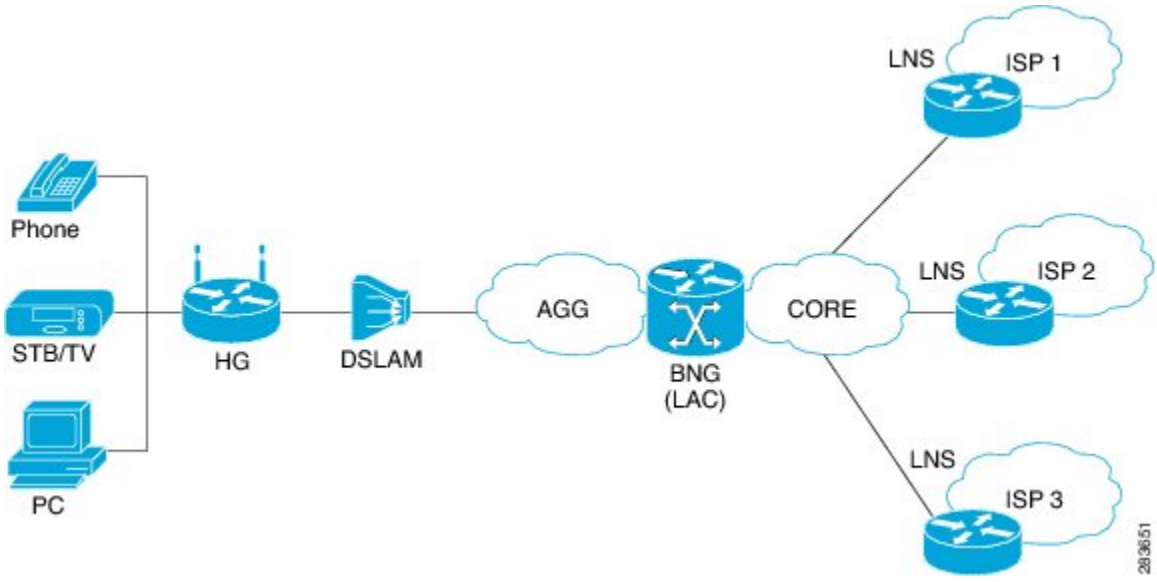


ネットワーク サービス プロバイダー モデルでは、ISP（小売業者とも呼ばれる）が加入者へのブロードバンド接続を直接提供します。 上の図に示すように、BNG はエッジ ルータにあり、その役割はアップリンクを介してコア ネットワークに接続することです。

アクセス ネットワーク プロバイダー

次の図は、アクセス ネットワーク プロバイダー モデルのトポロジを示しています。

図 *3*：アクセス ネットワーク プロバイダー モデル



アクセス ネットワーク プロバイダー モデルでは、ネットワーク事業者（卸売業者とも呼ばれる）がエッジ ネットワーク インフラストラクチャを所有し、加入者へのブロードバンド接続を提供し

ます。 ただし、ネットワーク事業者はブロードバンド ネットワークを所有していません。 その代わり、ネットワーク事業者はブロードバンドネットワークを管理する ISP の 1 つに接続します。

BNG はネットワーク事業者によって実装され、その役割は複数の ISP の 1 つに加入者トラフィックを渡すことです。 事業者から ISP へのハンドオフ タスクは、レイヤ 2 トンネリング プロトコル（L2TP）またはレイヤ 3 仮想プライベート ネットワーキング（VPN）によって実装されます。 L2TP には、次の 2 つの異なるネットワーク コンポーネントが必要です。

- L2TP アクセス コンセントレータ（LAC）：LAC は BNG によって提供されます。
- L2TP ネットワーク サーバ（LNS）：LNS は ISP によって提供されます。

BNG パッケージ

BNG フィーチャ セットは、プラットフォームに依存しない（PI）コンポーネントとプラットフォームに依存する（PD）コンポーネントの両方を含む複数のコンポーネントによって提供されます。 BNG パッケージは、BNG フィーチャ セットを提供するコンポーネントの集まりです。 BNG は、ASR9K 導入のサブセットでのみ使用されます。 その結果、BNG パッケージ機能によって、BNG 機能を使用しないシステムで最小のシステム リソースが使用されます。

BNG フィーチャ セットをサポートする BNG PIE は、ASR9k プラットフォームでコンパイル、インストール、アンインストール、アクティブ化、および非アクティブ化される場合があります。 BNG PIE の要件は、次のとおりです。

- BNG フィーチャ セットは、BNG PIE がインストールされるまで ASR9K で使用できません。
- ASR9K CPU は、BNG PIE がインストールされるときに再起動されません。
- 関連する BNG の設定は、BNG PIE が削除または非アクティブ化されるときに実行コンフィギュレーションから削除される必要があります。

BNG 機能は、x86 ベースのプラットフォームでのみ導入され、使用可能な PIE は **asr9k-bng-px.pie** です。

Cisco ASR 9000 ルータでの BNG PIE の構築およびインストール

Cisco ASR 9000 ルータで BNG PIE をインストールするには、次の作業を実行します。

はじめる前に

x86 ベースのルート スイッチ プロセッサ（RSP）に対して、**jam asr9k-bng-px.pie** コマンドを使用して BNG PIE を構築します。

手順の概要

1. **admin**
2. **install add** {*pie_location* | **source** | **tar**}
3. **install activate** {*pie_name* | **id**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**admin** 例： `RP/0/RSP0/CPU0:router# admin`	管理モードを開始します。
ステップ 2	**install add** {*pie_location* | **source** | **tar**} 例： `RP/0/RSP0/CPU0:router(admin)# install add /asr9k-bng-px.pie`	Cisco ASR 9000 ルータで、PIE ファイルをインストールします。
ステップ 3	**install activate** {*pie_name* | **id**} 例： `RP/0/RSP0/CPU0:router(admin)# install activate asr9k-bng-px.pie`	Cisco ASR 9000 ルータで、インストールされた PIE をアクティブ化します。

次の作業

（注） 421 から 430 へのアップグレード中、BNG PIE（asr9k-bng-px.pie）をインストールする前に ASR9k ベース イメージの PIE（asr9k-mini-px.pie）をインストールすることを推奨します。

BNG PIE をインストールしたら、フラッシュまたは tftp ロケーションから BNG 関連の設定をコピーします。 BNG PIE が非アクティブ化されて再度アクティブ化された場合、設定端末から **load config removed** コマンドを実行して、削除された BNG の設定をコピーします。

（注） ほとんどの BNG の機能の設定は、新しい名前空間のパーティションに移動され、BNG 機能はデフォルトでは使用できません。 これは、BNG PIE のインストール前後でいくつかのコマンドが一致しないことを意味します。 そのため、必要に応じて、「clear configs inconsistency」を実行する必要がある場合があります。

BNG 設定プロセス

Cisco ASR 9000 シリーズ ルータでの BNG の設定には、次の段階があります。

- RADIUS サーバの設定：BNG は、認証、許可、アカウンティング機能について RADIUS サーバと対話するように設定されます。 詳細については、認証、許可、アカウンティング機能の設定，（25 ページ）を参照してください。
- コントロール ポリシーのアクティブ化：コントロール ポリシーをアクティブ化し、特定のイベントが発生したときに BNG が実行するアクションを決定します。 アクションの手順は、ポリシー マップに示されています。 詳細については、コントロール ポリシーのアクティブ化，（77 ページ）を参照してください。
- 加入者セッションの確立：ブロードバンド サービスへのアクセスについて、加入者からネットワークに 1 つまたは複数の論理セッションを設定します。 各セッションは、一意に追跡および管理されます。 詳細については、加入者セッションの確立，（91 ページ）を参照してください。
- QoS の導入：Quality of Service（QoS）は、さまざまなネットワーク アプリケーションおよびトラフィック タイプを制御するために導入されます。 たとえば、サービス プロバイダーは、各加入者に割り当てられるリソース（サンプル帯域幅）を制御し、カスタマイズされたサービスを提供し、ミッションクリティカルなアプリケーションに属するトラフィックを優先させることができます。 詳細については、Quality of Service（QoS）の導入，（221 ページ）を参照してください。
- 加入者機能の設定：ポリシー ベース ルーティングなどの追加機能、アクセス リストやアクセス グループを使用したアクセス コントロール、およびマルチキャスト サービスを提供する特定の加入者機能をアクティブ化する設定を行います。 詳細については、加入者機能の設定，（269 ページ）を参照してください。
- セッション確立の確認：確立されたセッションは、接続が常に使用可能であることを保障するために確認およびモニタされます。 確認は、主に「show」コマンドを使用して行われます。 さまざまな「show」コマンドのリストについては、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』を参照してください。

BNG コマンドを使用するには、適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。 各コマンドに必要なタスク ID については、『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*』を参照してください。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

制約事項

BNG が設定されている場合は、選択的 VRF ダウンロード（SVD）をディセーブルにする必要があります。 SVD の詳細については、『*Cisco IOS XR Routing Configuration Guide for the Cisco XR 12000 Series Router*』を参照してください。

BNG のハードウェア要件

BNG をサポートするハードウェアは、次のとおりです。

- BNG は、衛星ネットワーク仮想化（nV）システムでサポートされます。

- BNG は、Cisco ASR 9922 シリーズ アグリゲーション サービス ルータでサポートされます。
- BNG は、RSP-440 ルート スイッチ プロセッサを搭載する Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでサポートされます。 RSP 2 ルート スイッチ プロセッサは、BNG をサポートしません。

表 *2* : *BNG* でサポートされるラインカードおよびモジュラ ポート アダプタ

ラインカード	モジュラ ポート アダプタ
24 ポート 10 ギガビット イーサネット ライン カード、最適化されたサービス エッジ	A9K-24X10GE-SE
36 ポート 10 ギガビット イーサネット ライン カード、最適化されたサービス エッジ	A9K-36X10GE-SE
80 ギガバイト モジュラ ラインカード、最適化されたサービス エッジ	A9K-MOD80-SE
160 ギガバイト モジュラ ラインカード、最適化されたサービス エッジ	A9K-MOD160-SE
20 ポート ギガビット イーサネット モジュラ ポート アダプタ（MPA）	A9K-MPA-20GE
2 ポート 10 ギガビット イーサネット モジュラ ポート アダプタ（MPA）	A9K-MPA-2X10GE
4 ポート 10 ギガビット イーサネット モジュラ ポート アダプタ（MPA）	A9K-MPA-4X10GE
2 ポート 40 ギガビット イーサネット モジュラ ポート アダプタ（MPA）	A9K-MPA-2X40GE
1 ポート 40 ギガビット イーサネット モジュラ ポート アダプタ（MPA）	A9K-MPA-1X40GE

BNG の相互運用性

BNG の相互運用性によって、BNG は他の大規模な異種ネットワークと情報を交換し、使用できるようになります。 主要な機能は、次のとおりです。

- BNG と ASR9001 の共存

ASR9001は、ルート スイッチプロセッサ（RSP）、ラインカード（LC）、およびイーサネット プラグ（EP）で構成される処理能力に優れたスタンドアロン ルータです。 すべての BNG 機能は、ASR9001 シャーシで完全にサポートされます。

- BNG による nV 衛星のサポート

  nV 衛星がサポートする 2 つのトポロジは、次のとおりです。

  - 単一のイーサネット ポート接続を介して ASR9K に接続される衛星ノードの CPE 側にあるバンドルされたイーサネット ポート。
  - 衛星からアクセス ネットワークへの非バンドル ポート、および衛星ノードと ASR9K 間のバンドル ポート。

- BNG とキャリア グレード NAT（CGN）の相互運用

  IPv4 アドレス空間の枯渇による差し迫った脅威に対処するために、残りの IPv4 アドレスまたは使用可能な IPv4 アドレスをより多くのカスタマー間で共有することを推奨します。 これは、サービス プロバイダー ネットワークのより集中型の NAT へのアドレス割り当てを主に実行する CGN を使用して行われます。 NAT44 は、CGN を使用するテクノロジーで、IPv4 アドレス空間の枯渇問題の管理に役立ちます。 BNG は、IPoE および PPPoE ベースの BNG 加入者セッションで NAT44 変換を実行する機能をサポートします。

制約事項

- 次のトポロジは、nV 衛星でサポートされません。
  - 単一のイーサネット ポート接続を介して ASR9K に接続される、衛星ノードの CPE 側にある単一のイーサネット ポート（非バンドル）。
  - バンドル イーサネット接続を介して ASR9K に接続される、衛星ノードの CPE 側にあるバンドルされたイーサネット ポート。
- BNG は、非バンドル ICL を持つ衛星ではサポートされません。

第 3 章

認証、許可、アカウンティング機能の設定

この章では、BNG ルータでの認証、許可、アカウンティング（AAA）機能の設定に関する情報を提供します。 BNG は、RADIUS サーバと対話して AAA 機能を実行します。 RADIUS サーバ グループは、特定の AAA タスクが割り当てられているサーバ グループを形成します。 サーバまたはサーバ グループで定義された方式リストには、許可が実行される方式が一覧表示されています。 RADIUS 機能の一部には、特定の AAA 属性形式の作成、RADIUS サーバのロード バランシング、RADIUS レコードのスロットリング、許可変更（CoA）、および QoS のサービス アカウンティングが含まれています。 この章の内容は、次のとおりです。

AAA の概要

AAA は、効果的なネットワーク管理およびセキュリティのフレームワークとして機能します。 これは、ネットワーク リソースの管理、ポリシーの施行、ネットワーク使用状況の監査、および課金関連情報の提供に役立ちます。 BNG は、AAA 機能を提供する外部の RADIUS サーバに接続します。

RADIUS サーバは、3 種類の独立したセキュリティ機能（認証、許可、アカウンティング）を実行して、不正アクセスからネットワークを保護します。 RADIUS サーバは、リモート認証ダイヤルイン ユーザ サービス（RADIUS）プロトコルを実行します。（RADIUS プロトコルの詳細については、RFC 2865 を参照してください）。 RADIUS サーバは、BNG、およびユーザ情報を含むデータベースとディレクトリと対話することによって AAA プロセスを管理します。

RADIUS プロトコルは、分散型クライアント/サーバ システムで動作します。 RADIUS クライアントは、中央の RADIUS サーバに認証要求を送信する BNG（Cisco ASR 9000 シリーズ ルータ）で実行されます。 RADIUS サーバには、すべてのユーザ認証情報とネットワーク サービス アクセス情報が含まれています。

AAA プロセス、これらのプロセス中の RADIUS サーバの役割、および一部の BNG の制約事項については、次の項で説明します。

認証

認証プロセスは、ネットワークおよびネットワーク サービスへのアクセスを許可する前に、ネットワーク上の加入者を識別します。 認証プロセスは、ネットワークへのアクセス権を取得するために各加入者が持つ一意の基準セットで機能します。 通常、RADIUS サーバは、加入者がその加入者のデータベースに入力したクレデンシャル（ユーザ名およびパスワード）を照合することによって認証を実行します。 クレデンシャルが一致した場合、加入者はネットワークへのアクセスが許可されます。 それ以外の場合は、認証プロセスが失敗し、ネットワークへのアクセスは拒否されます。

許可

認証プロセスの後、加入者は特定のアクティビティを実行することが許可されます。 許可は、加入者が使用を許可されるアクティビティ、リソース、またはサービスの種類を決定するプロセスです。 たとえば、ネットワークにログインした後に、加入者は、データベースまたは制限された Web サイトにアクセスしようとすることがあります。 許可プロセスでは、加入者がこれらのネットワーク リソースにアクセスする権限があるかどうかが判断されます。

AAA 許可は、加入者が提供する認証クレデンシャルに基づいて一連の属性を組み合わせて機能します。 RADIUS サーバは、指定のユーザ名について、これらの属性とデータベースに格納されている情報を比較します。 その加入者に適用されている実際の機能と制限事項を判断するための結果が BNG に返されます。

アカウンティング

アカウンティングは、ネットワーク アクセス中に加入者が使用するリソースを追跡します。アカウンティングは、課金、トレンド分析、リソース使用率の追跡、およびキャパシティ プランニング アクティビティに使用されます。 アカウンティング プロセス中、ログはネットワーク使用統計情報について保持されます。 モニタされる情報には、加入者 ID、加入者に適用されている設定、ネットワーク接続の開始時刻と終了時刻、およびネットワークとの間で転送されたパケット数とバイト数が含まれますが、これに限定されません。

BNG は、アカウンティング レコードの形式で RADIUS サーバに加入者アクティビティを報告します。 各アカウンティング レコードは、アカウンティング属性値で構成されます。 この値は、

ネットワーク管理、クライアント課金、監査などに対して、RADIUS サーバによって分析され、使用されます。

加入者セッションのアカウンティング レコードは、BNG が RADIUS サーバから応答を受信しなければタイムアウトすることがあります。 このタイムアウトは、到達不能の RADIUS サーバまたは RADIUS サーバのパフォーマンスの低下につながるネットワーク接続の問題に原因がある可能性があります。 BNG でのセッションがアカウント開始要求について承認されなければ、ルート プロセッサ フェールオーバー（RPFO）でのセッションの切断とその他の重大な障害が報告されます。 そのため、セッションの切断を避けるために、BNG で RADIUS サーバの**デッド タイム**を設定することを推奨します。 この値が設定され、再試行後にも特定のセッションがアカウンティング応答を受信していない場合、特定の RADIUS サーバは機能していないと考えられ、以降の要求はそのサーバに送信されません。

radius-server deadtime *limit* コマンドを使用して、RADIUS サーバの**デッド タイム**を設定できます。 詳細については、RADIUS サーバの設定, （48 ページ）を参照してください。

制約事項

- BNG では、ローカル認証とローカル許可はサポートされません。 RADIUS サーバによって実行される必要があります。
- セッションの接続が解除されると、システムがハードウェアから収集する「最後の」セッション統計情報を待つ間、RADIUS へのアカウンティング停止要求の送信が数秒間遅れることがあります。 ただし、アカウンティング停止要求の Event-Timestamp 属性は、転送時間ではなくクライアントが接続解除した時間を反映します。

RADIUS サーバ グループの使用

RADIUS サーバ グループは、1 つまたは複数の RADIUS サーバの名前付きグループです。 各サーバ グループは、特定のサービスに使用されます。 たとえば、2 つの RADIUS サーバ グループがある AAA ネットワーク設定では、最初のサーバ グループに認証タスクと許可タスクを割り当てることができ、2 番目のグループにアカウンティング タスクを割り当てることができます。

サーバ グループは、同じサーバに複数のホスト エントリを含めることができます。 ただし、各エントリに固有識別子が必要です。 この固有識別子は、IP アドレスと UDP ポート番号の組み合わせによって作成されます。 そのため、サーバの異なるポートを、特定の AAA サービスを提供する個別の RADIUS ホストとして別々に定義できます。 つまり、この固有識別子によって、同じサーバ上の異なる UDP ポートに RADIUS 要求を送信できます。 さらに、同じ RADIUS サーバ上の異なる 2 つのホスト エントリが同じサービス（認証プロセスなど）に対して設定されている場合、2 番目のホスト エントリは最初のホスト エントリのフェールオーバー バックアップとして機能します。 つまり、最初のホスト エントリが認証サービスの提供に失敗した場合、BNG は 2 番目のホスト エントリで試みます。 （RADIUS ホスト エントリは、それらが作成された順番で試行されます）。

サーバ グループへの特定のアクションの割り当てについては、RADIUS サーバ グループの設定, （28 ページ）を参照してください。

RADIUS サーバ グループの設定

名前付きサーバ グループをサーバ ホストとして定義するには、この作業を実行します。

手順の概要

1. **configure**
2. **aaa group server radius** *name*
3. **accounting accept** *radius_attribute_list_name*
4. **authorization reply accept** *radius_attribute_list_name*
5. **deadtime** *limit*
6. **load-balance method least-outstanding batch-size** *size* **ignore-preferred-server**
7. **server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*
8. **source-interface** *name value*
9. **vrf** *name*
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa group server radius** *name* 例： `RP/0/RSP0/CPU0:router(config)# aaa group server radius r1`	r1 という名前の RADIUS サーバ グループを設定します。
ステップ 3	**accounting accept** *radius_attribute_list_name* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# accounting accept att_list`	リストに指定されている属性のみを受け入れるように、アカウンティング プロセスの RADIUS 属性フィルタを設定します。

	コマンドまたはアクション	目的
ステップ 4	**authorization reply accept** *radius_attribute_list_name* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# authorization reply accept att_list1`	リストに指定されている属性のみを受け入れるように、許可プロセスのRADIUS属性フィルタを設定します。
ステップ 5	**deadtime** *limit* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# deadtime 40`	RADIUS サーバ グループのデッドタイムを設定します。 デッドタイムの制限は、分単位で設定します。 指定できる範囲は1～1440で、デフォルトは 0 です。
ステップ 6	**load-balance method least-outstanding batch-size** *size* **ignore-preferred-server** 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# load-balance method least-outstanding batch-size 50 ignore-preferred-server`	次のホストが選択されるまでのロード バランシングのバッチ サイズを設定します。
ステップ 7	**server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# server 1.2.3.4 acct-port 455 auth-port 567`	RADIUS サーバとそのホスト名を指定します。 RADIUS アカウンティングおよび認証要求の UDP ポートを設定します。 アカウンティングおよび認証ポート番号の範囲は、0～65535 です。 値が指定されていない場合、認証ポートのデフォルトは 1645、アカウンティング ポートのデフォルトは 1646 です。
ステップ 8	**source-interface** *name value* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# source-interface Bundle-Ether 455`	RADIUS サーバ グループの送信元インターフェイス名と Bundle-Ether の値を設定します。
ステップ 9	**vrf** *name* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# vrf vrf_1`	サーバの RADIUS グループが属する VRF を指定します。
ステップ 10	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS サーバ グループの設定：例

```
configure
aaa group server radius r1
accounting accept r1 r2
authorization reply accept a1 a2
deadtime 8
load-balance method least-outstanding batch-size 45 ignore-preferred-server
server host_name acct-port 355 auth-port 544
source-interface Bundle-Ether100.10
vrf vrf_1
!
end
```

方式リストの指定

AAAの方式リストは、許可が実行される方式、およびこれらの方式が実行される順序を定義します。 定義された認証方式が実行される前に、ユーザアクセスクレデンシャルの検証を実行する設定メカニズムに方式リストを適用する必要があります。 この要件の唯一の例外は、デフォルトの方式リスト（「default」という名前）です。 デフォルトの方式リストは、他の方式リストが定義

されていない場合に自動的に適用されます。 定義済みの方式リストは、デフォルトの方式リストに優先します。

BNG では、方式リスト、および AAA サービスに使用されるサーバ グループを指定する必要があります。 方式リストの指定については、AAA の方式リストの設定, （31 ページ）を参照してください。

AAA の方式リストの設定

加入者の認証、許可、アカウンティングについてサーバ グループが使用する方式リストを割り当てるには、次の作業を実行します。

手順の概要

1. **configure**
2. **aaa authentication subscriber default** *method-list-name* **group** *server-group-name*
3. **aaa authorization subscriber default** *method-list-name* **group** *server-group-name* |**radius**
4. **aaa accounting subscriber default** *method-list-name* **group** *server-group-name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa authentication subscriber default** *method-list-name* **group** *server-group-name* 例： `RP/0/RSP0/CPU0:router(config)# aaa authentication subscriber default method1 group group1 radius group group2 group group3 ...`	加入者の認証についてデフォルトで適用される方式リストを設定します。 「デフォルト」または AAA 方式リストのユーザ定義名のいずれかを入力できます。 また、方式リストを適用するサーバ グループの名前も入力します。
ステップ 3	**aaa authorization subscriber default** *method-list-name* **group** *server-group-name* |**radius**	加入者の許可についてデフォルトで適用される方式リストを設定します。 「デフォルト」または AAA 方式リストのユーザ定義名のいずれかを入力できます。 また、方式リストを適用するサーバ グループの名前も入力します。

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router(config)# aaa authorization subscriber default method1 group group1 radius group group2 group group3 ...`	
ステップ 4	**aaa accounting subscriber default** *method-list-name* **group** *server-group-name* 例： `RP/0/RSP0/CPU0:router(config)# aaa accounting subscriber default method1 group group1 radius group group2 group group3 ...`	加入者のアカウンティングについてデフォルトで適用される方式リストを設定します。「デフォルト」または AAA 方式リストのユーザ定義名のいずれかを入力できます。また、方式リストを適用するサーバ グループの名前も入力します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

AAA の方式リストの設定：例

```
configure
aaa authentication subscriber default group radius group rad2 group rad3..
```

```
aaa authorization subscriber default group radius group rad1 group rad2 group rad3..
aaa accounting subscriber default group radius group rad1 group rad2 group rad3..
!
!
end
```

AAA 属性の定義

AAA 属性は、RAIDUS パケットの要素です。 RAIDUS パケットは、RADIUS サーバと RADIUS クライアント間でデータを転送します。 AAA 属性パラメータとその値は、属性値ペア（AVP）を形成します。 AVP は、AAA トランザクションの要求と応答の両方に対してデータを伝送します。

AAA 属性は、インターネット技術特別調査委員会（IETF）属性などで事前定義されるか、ベンダー固有属性（VSA）などでベンダー定義されます。 BNG のサポートされる属性のリストの詳細については、RADIUS 属性, （367 ページ）を参照してください。

RADIUS サーバは、RADIUS メッセージの属性の形式で、BNG への設定の更新を提供します。 設定の更新は、2 種類の典型的な方式でのセッションのセットアップ中に加入者に適用されます。この方式とは、ユーザ単位の属性で、加入者の認証 Access Accept の一部として、または明示的ドメイン、ポート、またはサービスの許可 Access Accept を介して加入者に設定を適用します。 これは、加入者のポリシー ルール エンジンの設定によって完全に制御されます。

BNG がアクセス要求として外部の RADIUS サーバに認証要求または許可要求を送信すると、サーバは Access Accept の一部として BNG に設定の更新を送り返します。 セットアップ中に加入者を設定する RADIUS に加えて、BNG が要求を送信できなかった場合でも、サーバは加入者のアクティブなセッションのライフサイクル中に、許可変更（CoA）メッセージを自律的に BNG に送信できます。 これらの RADIUS CoA の更新は、BNG で設定された要素を参照し、特定のコントロール ポリシーまたはサービス ポリシーを更新するように BNG に指示する動的な更新として機能します。

BNG は、そのサービスを表すために共同作業できる設定済み機能グループである「サービス」の概念をサポートします。 サービスは、CLI を使用して動的なテンプレートに設定されている機能、または RADIUS サーバ内の RADIUS 属性として設定されている機能のいずれかとして表すことができます。 サービスは、ポリシー ルール エンジンの設定済み「アクティブ化」アクションまたは CoA の「アクティブ化サービス」要求のいずれかを介して、CLI または RADIUS から直接的にアクティブ化されます。 サービスは、ポリシー ルール エンジンの設定済み「非アクティブ化」アクションまたは CoA の「非アクティブ化サービス」要求を介して、直接的に非アクティブ化する（名前付きサービス内の関連機能をすべて削除する）こともできます。

RADIUS から受信した属性値は、次の方法で加入者セッションと対話します。

- BNG は、CLI コマンドによって静的にプロビジョニングされた既存の値とともに RADIUS の更新で受信した値、または以前の RADIUS の更新から受信した値をマージします。
- いずれの場合も、RADIUS の更新で受信した値は、対応する CLI のプロビジョニングされた値または以前の RADIUS の更新に優先します。 CLI のプロビジョニングされた値を再設定していても、システムは RADIUS の更新で受信されたセッション属性や機能を上書きしません。
- 動的なテンプレートでの CLI のプロビジョニング値に対する変更は、テンプレート機能が RADIUS によってすでに上書きされていないことを前提として、そのテンプレートを使用す

るすべてのセッションですぐに有効になります。 同様に、CoAの「サービス更新」要求によるサービスの更新に適用されます。

AAA 属性リスト

属性リストは、一連の属性を含む名前付きリストです。 AAA機能を実行するために特定の属性のリストを使用するようにRADIUSサーバを設定できます。

属性リストを作成するには、RADIUS 属性リストの設定, （40 ページ）を参照してください。

AAA 属性形式

一部の属性のカスタマイズされた形式を定義できます。 新しい形式を作成するための設定構文は、次のとおりです。

```
aaa attribute format FORMAT-NAME format-string [LENGTH] STRING *[Identity-Attribute]
```

値は次のとおりです。

- FORMAT-NAME：属性形式に割り当てる名前を指定します。 この名前は、形式が属性に適用されるときに参照されます。
- LENGTH：（任意）フォーマットされた属性文字列の最大長を指定します。 属性文字列の最後の長さがLENGTHで指定した値を超えると、LENGTHバイトに丸められます。 LENGTHに許容される最大値は255です。 引数が設定されていない場合は、デフォルトも255です。
- STRING：変換指定子を含む通常のASCII文字を含みます。 %記号のみ、STRINGで変換指定子として許容されます。 STRING値は、二重引用符で囲まれます。
- Identity-Attribute：セッションを識別し、ユーザ名、IPアドレス、およびMACアドレスが含まれます。 現在定義されているID属性のリストは、CLIに表示されます。

形式が定義されると、ユーザ名、Nas-Port-ID、Calling-Station-ID、およびCalled-Station-IDなど、さまざまなAAA属性にFORMAT-NAMEを適用できます。 形式の機能を使用する設定可能なAAA属性については、特定形式の属性の作成, （34 ページ）の項で説明します。

カスタマイズされたNas-Port属性を作成し、事前定義された形式をNas-Port-ID属性に適用するには、RADIUS 属性形式の設定, （42 ページ）を参照してください。

特定の目的に対する属性形式に、特定の機能を定義できます。 たとえば、入力ユーザ名が「text@abc.com」で、「@」の後の部分のみがユーザ名として必要な場合、機能を定義して、「@」の後の部分のみをユーザ名として保持できます。 「text」が入力からドロップされ、新しいユーザ名は「abc.com」になります。 ユーザ名のトランケーション機能を名前付き属性形式に適用するには、AAA 属性形式機能の設定, （45 ページ）を参照してください。

特定形式の属性の作成

BNGは、設定可能なAAA属性の使用をサポートします。 設定可能なAAA属性には、特定のユーザ定義の形式があります。 ここでは、BNGで使用される設定可能なAAA属性の一部を示します。

Username

BNG には、MAC アドレス、Circuit-ID、Remote-ID、および DHCP オプション 60（および CLI で使用できる多数の値セット）を使用する加入者の AAA ユーザ名とその他の形式がサポートされた属性を構築する機能があります。 DHCP オプション 60 は、要求に応じて DHCP クライアントから DHCP サーバに伝送される新しいオプションの 1 つです。この機能は、DHCP クライアントハードウェアのベンダー クラス ID（VCI）を伝送します。

MAC アドレス属性は、次のいずれかの形式の CLI 形式で指定されます。

- mac-address：0000.4096.3e4a など
- mac-address-ietf：00-00-40-96-3E-4A など
- mac-address-raw：000040963e4a など

「mac-address@vendor-class-ID」形式でのユーザ名の作成例は、次のとおりです。

```
aaa attribute format USERNAME-FORMAT format-string "%s@%s" mac-address vendor-class-id
```

NAS-Port-ID

NAS-Port-ID は、BNG ポート情報とアクセス ノード情報の組み合わせによって構築されます。BNG ポート情報は、次の形式の文字列で構成されます。

```
"eth phy_slot/phy_subslot/phy_port:XPI.XCI"
```

802.1Q トンネリング（QinQ）では、XPI は外部 VLAN タグで、XCI は内部 VLAN タグです。

インターフェイスが QinQ の場合、Nas-Port-ID のデフォルト形式には両方の VLAN タグが含まれます。インターフェイスがシングル タグの場合、単一の VLAN タグが含まれます。

単一の VLAN の場合、次の構文を使用して外部 VLAN のみが設定されます。

```
<slot>/<subslot>/<port>/<outer_vlan>
```

QinQ の場合、次の構文を使用して VLAN が設定されます。

```
<slot>/<subslot>/<port>/<inner_vlan>.<outer_vlan>
```

Nas-Port-ID コマンドは、（前述のコマンドを使用して設定された）カスタマイズされた形式を特定のインターフェイス タイプ（NAS-Port-Type）で使用できるように、「NAS-Port-Type」オプションを使用するように拡張されています。 拡張された Nas-Port-ID コマンドは、次のとおりです。

```
aaa radius attribute nas-port-id format FORMAT_NAME [type NAS_PORT_TYPE]
```

「Type」オプションを指定しないと、すべてのインターフェイス タイプの Nas-Port-ID がコマンドで指定されている形式名に従って構成されます。 BNG ポート情報と Circuit-ID を組み合わせることによって、最大 128 バイトの NAS-Port-ID を作成する例は、次のとおりです。

```
aaa attribute format NAS-PORT-ID-FORMAT1 format-string 128 "eth %s/%s/%s:%s.%s %s" phy-slot
 phy-subslot phy-port outer-vlan-Id
inner-vlan-id circuit-id
```

Circuit-ID の最後に「0/0/0/0/0/0」を追加して、単なる BNG ポート情報から NAS-Port-ID を作成する例は、次のとおりです。

```
aaa attribute format NAS-PORT-ID-FORMAT2 format-string "eth %s/%s/%s:%s.%s 0/0/0/0/0/0"
phy-slot phy-subslot phy-port outer-vlan-Id
inner-vlan-id
```

単なる Circuit-ID から NAS-Port-ID を作成する例は、次のとおりです。

```
aaa attribute format NAS-PORT-ID-FORMAT3 format-string "%s" circuit-id
```

前述の例で設定した NAS-Port-ID 形式は、次のように NAS-Port-ID コマンドで指定できます。

```
For  IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT1  type 41

For  Virtual IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT2 type 44

For  IPOEoE interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT3 type 39
```

インターフェイスまたは VLAN サブインターフェイスの NAS-Port-Type

同じタイプのさまざまな物理インターフェイスを除いて、同じ BNG ルータ上の加入者に対して異なる製造モデルを持つには、各物理インターフェイスまたは VLAN サブインターフェイスに対して NAS-Port-Type を設定可能にします。 インターフェイス上で設定された異なる NAS-Port-Type 値によって、NAS-Port と NAS-Port-ID は、インターフェイスにある NAS-Port-Type の実際の値ではなく、インターフェイス上で設定された新しい NAS-Port-Type にグローバルに定義された形式に従って形式作成されます。 これにより、NAS-Port、NAS-Port-ID、および NAS-Port-Type の異なる形式が、異なる製造モデル下の加入者の RADIUS サーバに順番に送信されます。

サブインターフェイスの場合、RADIUS サーバに送信される NAS-Port-Type 形式の決定で従うべき階層は、次のとおりです。

1. NAS-Port-Type が加入者セッションが到着するサブインターフェイスで設定されているかどうかを確認します。
2. NAS-Port-Type がサブインターフェイスで設定されていない場合は、メインの物理インターフェイスで設定されているかどうかを確認します。

   NAS-Port または NAS-Port-ID の形式は、手順 1 または手順 2 で取得される NAS-Port-Type に基づきます。
3. NAS-Port-Type がサブインターフェイスとメインの物理インターフェイスのいずれでも設定されていない場合、NAS-Port または NAS-Port-ID の形式は、サブインターフェイスのデフォルトの NAS-Port-Type 形式に基づきます。
4. NAS-Port または NAS-Port-ID 形式が手順 1、2 または 3 で取得される NAS-Port-Type に設定されていない場合、NAS-Port または NAS-Port-ID の形式は、NAS-Port または NAS-Port-ID のデフォルト形式に基づきます。

インターフェイスまたは VLAN サブインターフェイスごとに NAS-Port-Type を設定するには、次のコマンドを使用します。

```
aaa radius attribute nas-port-type <nas-port-type>
```

値は次のとおりです。

<nas-port-type> は、0 ～ 44 の範囲または NAS-Port-Type を指定した文字列のいずれかになります。

RADIUS 属性の NAS-Port-Type の設定, （44 ページ）を参照してください。

Calling-Station-ID および Called-Station-ID

BNG は、設定可能な Calling-Station-ID および Called-Station-ID の使用をサポートします。 Calling-Station-ID は、自動番号識別（ANI）または同様のテクノロジーを使用する RADIUS 属性

です。 これにより、ネットワーク アクセス サーバ（NAS）はアクセス要求パケットにコールが着信した電話番号を送信できます。 Called-Station-ID は、着信番号識別サービス（DNIS）または同様のテクノロジーを使用する RADIUS 属性です。 これにより、NAS は、アクセス要求パケットにユーザがコールした電話番号を送信できます。

Calling-Station-ID および Called-Station-ID 属性の設定に使用するコマンドは、次のとおりです。

```
aaa radius attribute calling-station-id format FORMAT_NAME

aaa radius attribute called-station-id format FORMAT_NAME
```

MAC アドレス、Remote-ID、および Circuit-ID から Calling-Station-ID を作成する例は、次のとおりです。

```
aaa radius attribute calling-station-id format CLID-FORMAT

aaa attribute format CLID-FORMAT format-string "%s:%s:%s" mac-address-ietf remote-id
circuit-id
```

MAC アドレス、Remote-ID、および Circuit-ID から Called-Station-ID を作成する例は、次のとおりです。

```
aaa radius attribute called-station-id format CLDID-FORMAT

aaa attribute format CLDID-FORMAT format-string "%s:%s" mac-address-raw circuit-id
```

NAS-Port 形式

NAS-Port は、ブロードバンド リモート アクセス サーバ（BRAS）の物理ポート情報を持つ 4 バイトの値で、アクセス集約ネットワークを BNG に接続します。 これは、アクセス要求パケットとアカウンティング要求パケットの両方で使用されます。 BRAS の物理ポートを一意に識別するために、シェルフ、スロット、アダプタなどの複数の情報がポート番号と一緒に使用されます。 format-e と呼ばれる設定可能な形式は、NAS-Port の 32 ビットの個々のビットまたはビット グループによって、ポート情報を構成するさまざまな部分を表現またはエンコードできるように定義されます。

NAS-Port の個々のビットは、次の文字でエンコードできます。

- ゼロ：0
- 1：1
- PPPoX スロット：S
- PPPoX アダプタ：A
- PPPoX ポート：P
- PPPoX VLAN ID：V
- PPPoX VPI：I
- PPPoX VCI：C
- セッション ID：U

- PPPoX 内部 VLAN ID：Q

```
aaa radius attribute nas-port format e [string] [type {nas-port-type}]
```

前述のコマンドは、NAS-Port-Type（RADIUS 属性 61）の特定インターフェイスに対して format-e のエンコード文字列を設定するために使用されます。 許容される NAS-Port-Type の値は、次のとおりです。

NAS-Port-Type	値	関連するインターフェイスから値を取得できるかどうか	インターフェイス コンフィギュレーション モードで値を設定できるかどうか
ASYNC	0	No	Yes
SYNC	1	No	Yes
ISDN	2	No	Yes
ISDN_V120	3	No	Yes
ISDN_V110	4	No	Yes
VIRTUAL	5	No	Yes
ISDN_PIAFS	6	No	Yes
X75	9	No	Yes
ETHERNET	15	No	Yes
PPPATM	30	No	Yes
PPPOEOA	31	No	Yes
PPPOEOE	32	Yes	Yes
PPPOEOVLAN	33	Yes	Yes
PPPOEOQINQ	34	Yes	Yes
VIRTUAL_PPPOEOE	35	Yes	Yes
VIRTUAL_PPPOEOVLAN	36	Yes	Yes
VIRTUAL_PPPOEOQINQ	37	Yes	Yes
IPSEC	38	No	Yes
IPOEOE	39	Yes	Yes
IPOEOVLAN	40	Yes	Yes

NAS-Port-Type	値	関連するインターフェイスから値を取得できるかどうか	インターフェイス コンフィギュレーション モードで値を設定できるかどうか
IPOEOQINQ	41	Yes	Yes
VIRTUAL_IPOEOE	42	Yes	Yes
VIRTUAL_IPOEOVLAN	43	Yes	Yes
VIRTUAL_IPOEOQINQ	44	Yes	Yes

次に、例を示します。

```
For non-bundle: GigabitEthernet0/1/2/3.11.pppoe5

where:
PPPoEoQinQ (assuming 2 vlan tags): interface-type
1: slot
2: adapter
3: port
vlan-ids: whatever the outer and inner vlan-ids received in the PADR were
5: session-id

aaa radius attribute nas-port format e SSAAPPPPQQQQQQQQQQQVVVVVVVVVVVUUUU type 34
Generated NAS-Port:     01100011QQQQQQQQQQQVVVVVVVVVV0101

For bundle: Bundle-Ether17.23.pppoe8
where:
Virtual-PPPoEoQinQ (assuming 2 vlan tags): interface-type
0: slot
0: adapter
17 (bundle-id): port
Vlan-Ids: whatever the outer and inner vlan-ids received in the PADR were.
8: session-id

aaa radius attribute nas-port format e PPPPPPQQQQQQQQQQQVVVVVVVVVVUUUUUU type 37
Generated NAS-Port:     010001QQQQQQQQQQQVVVVVVVVV000101
```

IP/DHCP セッションの NAS-Port 形式を次の例に示します。

```
For IPoEoVLAN interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVVVV type 40

For IPoEoQinQ:
aaa radius attribute nas-port format e SSAAAPPPPPQQQQQQQQQQQQVVVVVVVVV type 41

For virtual IPoEoVLAN:
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVVUUUUUUUU type 43
```

PPPoE セッションの NAS-Port 形式を次の例に示します。

```
For PPPoEoVLAN  interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVUUUU type 33

For Virtual PPPoEoVLAN:.
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVVUUUUUUUU type 36
```

（注） NAS-Port 形式が NAS-Port-Type に対して設定されていない場合、システムは NAS-Port 形式のデフォルトの CLI 設定を検索します。 これらの両方の設定がない場合、特定の NAS-Port-Type を使用するセッションでは、NAS-Port 属性は RADIUS サーバに送信されません。

RADIUS 属性リストの設定

許可属性とアカウンティング属性のフィルタリングに使用される RADIUS 属性リストを作成するには、次の作業を実行します。

手順の概要

1. **configure**
2. **radius-server attribute list** *listname*
3. **attribute** *list_of_radius_attributes*
4. **attribute vendor-id** *vendor-type number*
5. **vendor-type** *vendor-type-value*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**radius-server attribute list** *listname* 例： `RP/0/RSP0/CPU0:router(config)# radius-server attribute list l1`	属性リストの名前を定義します。
ステップ 3	**attribute** *list_of_radius_attributes* 例： `RP/0/RSP0/CPU0:router(config-attribute-filter)# attribute a1, a2`	RADIUS 属性をリストに入力します。 （注） サポートされる属性の詳細については、RADIUS 属性, （367 ページ）を参照してください。

	コマンドまたはアクション	目的
ステップ 4	**attribute vendor-id** *vendor-type number* 例： `RP/0/RSP0/CPU0:router(config)# attribute vendor-id 6456`	ベンダー固有属性（VSA）に関するベンダー固有情報を RADIUS 属性リストの CLI で指定できるようにすることによって、VSA に適用される属性フィルタリングを設定します。 ベンダー固有情報は、シスコ汎用 VSA の場合、ベンダー ID、ベンダー タイプ、およびオプションの属性名で構成されます。 ベンダー ID の範囲は、0 ～ 4294967295 です。
ステップ 5	**vendor-type** *vendor-type-value* 例： `RP/0/RSP0/CPU0:router(config-attribute-filter-vsa)# vendor-type 54`	ベンダー タイプなどのベンダー固有情報を RADIUS 属性リストで指定されるように設定します。 ベンダー タイプ値の範囲は、1 ～ 254 です。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

RADIUS 属性リストの設定：例

```
configure
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
!
end
```

RADIUS 属性形式の設定

Nas-Port 属性の RADIUS 属性形式を定義し、Nas-Port-ID 属性に事前定義された形式を適用するには、次の作業を実行します。

手順の概要

1. **configure**
2. **aaa radius attribute**
3. **nas-port format e** *string* **type** *nas-port-type value*
4. **nas-port-id format** *format name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa radius attribute** 例： `RP/0/RSP0/CPU0:router(config)# aaa radius attribute`	AAA RADIUS 属性を設定します。

	コマンドまたはアクション	目的
ステップ 3	**nas-port format e** *string* **type** *nas-port-type value* 例： `RP/0/RSP0/CPU0:router(config)# nas-port format e format1 type 30`	Nas-Port 属性の形式を設定します。 文字列は、使用される形式を表す 32 文字の文字列を表します。 Nas-Port 値の範囲は 0～44 です。
ステップ 4	**nas-port-id format** *format name* 例： `RP/0/RSP0/CPU0:router(config)# nas-port-id format format2`	事前定義された形式を Nas-Port-ID 属性に適用します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS 属性形式の設定：例

```
configure
aaa radius attribute
nas-port format e abcd type 40
nas-port-id format ADEF
!
end
```

RADIUS 属性の NAS-Port-Type の設定

物理インターフェイスまたは VLAN サブインターフェイスの RADIUS 属性の NAS-Port-Type を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** *type interface-name*
3. **aaa radius attribute nas-port-type** {*value* | *name*}
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-name* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitEthernet 0/0/0/0`	インターフェイス コンフィギュレーション モードを開始します。

ステップ 3	**aaa radius attribute nas-port-type** {*value* | *name*}
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type 30`
	または
	`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type Ethernet`
		RADIUS 属性の NAS-Port-Type 値を設定します。
		この値の範囲は 0 ～ 44 です。
		この範囲内で許容される NAS-Port-Type 値については、NAS-Port 形式, （37 ページ）の表を参照してください。

ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS 属性の NAS-Port-Type の設定：例

```
configure
interface gigabitEthernet 0/0/0/0
 aaa radius attribute nas-port-type Ethernet
!
end
```

AAA 属性形式機能の設定

AAA 属性形式の機能を設定するには、次の作業を実行します。 この機能は、デリミタまでユーザ名を取り除くためのものです。

手順の概要

1. **configure**
2. **aaa attribute format** *format-name*
3. **username-strip prefix-delimiter** *prefilx_delimiter*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa attribute format** *format-name* 例： `RP/0/RSP0/CPU0:router(config)# aaa attribute format red`	機能が定義されている形式名を指定します。
ステップ 3	**username-strip prefix-delimiter** *prefilx_delimiter* 例： `RP/0/RSP0/CPU0:router(config-id-format)# username-strip prefix-delimiter @`	プレフィックスデリミタ（@）の前に付くユーザ名を取り除くように機能を設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

AAA 属性形式機能の設定：例

```
configure
aaa attribute format red
username-strip prefix-delimiter @
!
!
end
```

RADIUS サーバの設定

BNG が RADIUS サーバと対話するように、BNG ルータで特定のサーバ固有の設定を行う必要があります。 この表は、主要な設定の一部です。

設定	説明
サーバ ホスト	BNG が接続する RADIUS サーバの詳細を定義します。
属性リスト	使用される属性リストを定義します。
サーバ キー	暗号化ステータスを定義します。
デッド条件	RADIUS サーバをデッドとしてマークするために使用される条件を定義します。
再送信値	RADIUS サーバにデータを送信するために BNG が行う再試行数を定義します。
タイムアウトの値	BNG が RADIUS サーバの応答を待機する時間を定義します。
自動テスト	自動テストが開始されてからの期間とテストされるユーザ名を定義します。
IP DSCP	RADIUS パケットを特定の DiffServ コード ポイント（DSCP）値でマークできます。

RADIUS サーバ設定の詳細については、RADIUS サーバの設定, （48 ページ）を参照してください。

特定の自動テスト設定の詳細については、自動テストの設定, （52 ページ）を参照してください。

特定の IP DSCP 設定の詳細については、RADIUS サーバの IP DSCP の設定, （54 ページ）を参照してください。

制約事項

サービス プロファイルのプッシュまたはシステムへの非同期的なプロファイルのプッシュは、サポートされていません。 RADIUS からプロファイルをダウンロードするには、プロファイルを加

入者の要求の一部として最初に要求する必要があります。 サービスの更新のみがサポートされ、以前にダウンロードしたサービスの変更に使用できます。

RADIUS サーバの設定

BNG ルータで RADIUS サーバに固有の設定を行うには、次の作業を実行します。

手順の概要

1. **configure**
2. **radius-server host** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*
3. **radius-server attribute list** *list_name attribute_list*
4. **radius-server key** *7 encrypted_text*
5. **radius-server disallow null-username**
6. **radius-server dead-criteria time** *value*
7. **radius-server dead-criteria tries** *value*
8. **radius-server deadtime** *limit*
9. **radius-server ipv4 dscp** *codepoint_value*
10. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
11. **radius-server retransmit** *retransmit_value*
12. **radius-server source-port extended**
13. **radius-server timeout** *value*
14. **radius-server vsa attribute ignore unknown**
15. **radius source-interface Loopback** *value* **vrf** *vrf_name*
16. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**radius-server host** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number* 例： `RP/0/RSP0/CPU0:router(config)# radius-server host 1.2.3.4 acct-port 455 auth-port 567`	RADIUS サーバとそのホスト名を指定します。 RADIUS アカウンティングおよび認証要求の UDP ポートを設定します。 アカウンティングおよび認証ポート番号の範囲は、0 ～ 65535 です。 値が指定されていない場合、認証ポートのデフォルトは 1645、アカウンティング ポートのデフォルトは 1646 です。
ステップ 3	**radius-server attribute list** *list_name attribute_list* 例： `RP/0/RSP0/CPU0:router(config)# radius-server attribute list rad_list a b`	RADIUS サーバ属性リストを指定し、選択した RADIUS 属性をカスタマイズします。
ステップ 4	**radius-server key** *7 encrypted_text* 例： `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server key 7 rngiry`	デフォルトを上書きするサーバ単位の暗号キーを指定し、値 0 または 7 を取ります。これは、暗号化されていないキーが続くことを示します。
ステップ 5	**radius-server disallow null-username** 例： `RP/0/RSP0/CPU0:router(config)# radius-server disallow null-username`	RADIUS サーバにヌルユーザ名を許可しないように指定します。
ステップ 6	**radius-server dead-criteria time** *value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria time 40`	設定された RADIUS サーバのデッド サーバの検出基準を指定します。 時間（秒）は、この RADIUS サーバから応答を受信してから経過する必要のある最小時間を指定します。
ステップ 7	**radius-server dead-criteria tries** *value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria tries 50`	ルータで連続何回タイムアウトが発生したら、RADIUS サーバにデッドマークを付けるかを指定します。 値の範囲は 1 ～ 100 です。
ステップ 8	**radius-server deadtime** *limit* 例： `RP/0/RSP0/CPU0:router(config)# radius-server deadtime 67`	RADIUS サーバがデッドとマークされる時間を分単位で指定します。 デッドタイムの制限は、分単位で指定され、その範囲は 1 ～ 1440 です。 値が指定されていない場合のデフォルトは 0 です。

	コマンドまたはアクション	目的
ステップ 9	**radius-server ipv4 dscp** *codepoint_value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45`	RADIUS パケットを特定の DiffServ コードポイント（DSCP）値でマークできます。 このコード ポイント値の範囲は、0 ～ 63 です。
ステップ 10	**radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size* 例： `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500`	未処理トランザクションが最小のサーバを選択することによって、RADIUS ロードバランシング オプションを設定します。 このロードバランシング方式は、サーバの選択にバッチ サイズを使用します。 サイズの範囲は 1 ～ 1500 です。 値が指定されていない場合のデフォルトは 25 です。
ステップ 11	**radius-server retransmit** *retransmit_value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server retransmit 45`	アクティブ サーバへの再試行回数を指定します。 再送信値は、再試行数を数値で表し、その範囲は 1 ～ 100 です。 値が指定されていない場合のデフォルトは 3 です。
ステップ 12	**radius-server source-port extended** 例： `RP/0/RSP0/CPU0:router(config)# radius-server source-port extended`	BNG が RADIUS 要求を送信する送信元ポートとして合計 200 のポートを使用するように設定します。
ステップ 13	**radius-server timeout** *value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server timeout`	RADIUS サーバの応答を待機する時間を指定します。 値は秒単位で、その範囲は 1 ～ 1000 です。 デフォルト値は 5 です。
ステップ 14	**radius-server vsa attribute ignore unknown** 例： `RP/0/RSP0/CPU0:router(config)# radius-server vsa attribute ignore unknown`	RADIUS サーバの不明なベンダー固有属性を無視します。
ステップ 15	**radius source-interface Loopback** *value* **vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:router(config)# radius source-interface Loopback 655 vrf vrf_1`	RADIUS パケットの送信元アドレスにループバック インターフェイスを指定します。 値の範囲は 0 ～ 65535 です。
ステップ 16	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS サーバの設定：例

```
\\Configuring RADIUS Server Options
configure
radius-server attribute list list1 a b
radius-server dead-criteria time 100
radius-server deadtime 30
radius-server disallow null-username
radius-server host 1.2.3.4 acct-port 655 auth-port 566
radius-server ipv4 dscp 34
radius-server key 7 ERITY$
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 25
radius-server retransmit 50
radius-server source-port extended
radius-server timeout 500
radius-server vsa attribute ignore unknown
!
!
end
```

```
\\Configuring RADIUS Attribute List
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
!
end

\\Configuring RADIUS Server Host
configure
radius-server host 1.3.5.7 acct-port 56 auth-port 66
idle-time 45
ignore-acct-port
ignore-auth-port 3.4.5.6
key 7 ERWQ
retransmit 50
test username username
timeout 500
!
end

\\Configuring RADIUS Server Key
configure
radius-server key 7  ERWQ
!
end

\\Configuring Load Balancing for RADIUS Server
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end

\\Ignoring Unknown VSA Attributes in RADIUS Server
configure
radius-server vsa attribute ignore unknown
!
end

\\Configuring Dead Criteria for RADIUS Server
configure
radius-server dead-criteria time 60
radius-server dead-criteria tries 60
!
end

\\Configuring Disallow Username
configure
radius-server disallow null-username
!
end

\\Setting IP DSCP for RADIUS Server
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

自動テストの設定

外部 RADIUS サーバが UP かどうかをテストするには、次の作業を実行します。

手順の概要

1. **configure**
2. **radius-server idle-time** *idle_time*
3. **radius-server test username** *username*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**radius-server idle-time** *idle_time* 例： `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server idle-time 45`	自動テストが開始されるまでのアイドル時間を指定します。 アイドル時間は、分単位で指定され、その範囲は 1 ～ 60 です。
ステップ 3	**radius-server test username** *username* 例： `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server test username user1`	自動テスト機能用にテストされるユーザ名を指定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

自動テストの設定：例

```
configure
radius-server idle-time 60
radius-server test username user_1
!
end
```

RADIUS サーバの IP DSCP の設定

RADIUS サーバの IP DiffServ コードポイント（DSCP）を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **radius-server ipv4 dscp** *codepoint_value*
3. **radius-server ipv4 dscp default**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**radius-server ipv4 dscp** *codepoint_value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45`	RADIUS パケットを固有の DiffServ コード ポイント（DSCP）値でマークできます。この値は、最新の IP precedence、トラフィック タイプを分類して順位付けするために最初に使用される IP ヘッダーのタイプ オブ サービス バイトの 3 ビット フィールドを置き換えます。 このコード ポイント値の範囲は、0 ～ 63 です。
ステップ 3	**radius-server ipv4 dscp default** 例： `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp default`	パケットをデフォルトの DSCP（000000）と一致させます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS サーバの IP DSCP の設定：例

```
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

RADIUS サーバのトランザクション ロード バランシング

RADIUS ロードバランシング機能は、一連の RADIUS サーバに対する RADIUS アクセスおよびアカウンティング トランザクションの負荷を分担するメカニズムです。 各 AAA 要求の処理は、トランザクションと見なされます。 BNGは、トランザクションのバッチをサーバ グループ内のサーバに分配します。

新規に最初のトランザクションを受け取ると、BNGはキューの未処理のトランザクション数が最も少ないサーバを決定します。 このサーバは、そのトランザクションのバッチを割り当てられます。 BNGは、未処理のトランザクションが最も少ないサーバが常に新しいバッチを取得するように、この決定プロセスを繰り返し続けます。 この方法は、ロードバランシングの最小未処理方法として知られています。

ロード バランシング機能をグローバルに設定するか、サーバ グループに属する RADIUS サーバに対して設定できます。 サーバ グループでは、優先サーバが定義されている場合、ロードバランシング設定にキーワード「ignore-preferred-server」を含めてプリファレンスをディセーブルにする必要があります。

ロード バランシング機能をグローバルに設定する場合は、グローバル RADIUS サーバ グループのロード バランシングの設定, （56 ページ）を参照してください。

ロード バランシング機能を指定サーバ グループの一部である RADIUS サーバで設定するには、名前付き RADIUS サーバ グループのロード バランシングの設定, （58 ページ）を参照してください。

グローバル RADIUS サーバ グループのロード バランシングの設定

グローバル RADIUS サーバ グループのロード バランシング機能をアクティブ化するには、次の作業を実行します。 たとえば、この設定では、優先サーバを無視するように設定します。

手順の概要

1. **configure**
2. **radius-server load-balance method least-outstanding batch-size** *size*
3. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**radius-server load-balance method least-outstanding batch-size** *size* 例： `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding batch-size 500`	未処理トランザクションが最小のサーバを選択することによって、RADIUS ロードバランシング オプションを設定します。 このロードバランシング方式は、サーバの選択にバッチ サイズを使用します。 サイズの範囲は 1 ～ 1500 です。 値が指定されていない場合のデフォルトは 25 です。
ステップ 3	**radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size* 例： `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500`	このサーバ グループの優先サーバをディセーブルにして、RADIUS ロードバランシング オプションを設定します。 このロードバランシング方式は、サーバの選択にバッチ サイズを使用します。 サイズの範囲は 1 ～ 1500 です。 値が指定されていない場合のデフォルトは 25 です。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

RADIUS サーバのロード バランシングの設定：例

```
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end
```

名前付き RADIUS サーバ グループのロード バランシングの設定

名前付き RADIUS サーバ グループのロード バランシング機能をアクティブ化するには、次の作業を実行します。 たとえば、この設定では、優先サーバを無視するように設定します。

手順の概要

1. **configure**
2. **aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size*
3. **aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size* 例： `RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding batch-size 500`	未処理トランザクションが最小のサーバを選択することによって、RADIUS ロードバランシング オプションを設定します。 このロードバランシング方式は、サーバの選択にバッチ サイズを使用します。 サイズの範囲は 1 ～ 1500 です。 値が指定されていない場合のデフォルトは 25 です。
ステップ 3	**aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size* 例： `RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding ignore-preferred-server batch-size 500`	このサーバグループの優先サーバをディセーブルにして、RADIUS ロードバランシング オプションを設定します。 このロードバランシング方式は、サーバの選択にバッチ サイズを使用します。 サイズの範囲は 1 ～ 1500 です。 値が指定されていない場合のデフォルトは 25 です。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

RADIUS レコードのスロットリング

AAA（RADIUS）レコードのスロットリングは、RADIUS の輻輳と不安定性を防ぐメカニズムです。 この機能は、RADIUS サーバの BNG で生成される AAA 要求の突然のバーストに対応する帯域幅が不十分な場合に役立ちます。

スロットリングの設定時に、未処理要求の最大数に対応するしきい値レートが定義されます。 アクセス（認証および許可）およびアカウンティング要求に独立したスロットリング レートを設定できます。 しきい値がサーバに到達すると、そのタイプの要求はそれ以上サーバに送信されません。 ただし、保留中の要求については、再送信タイマーが開始され、（すべてのタイマーの期限切れ後にチェックされる）未処理要求の数がしきい値より少ない場合、要求が送信されます。

セッションはアクセス要求のスロットルが原因でタイムアウトすることがあるため、再送信の試行数に制限を設定します。 この制限に達すると、それ以上のアクセス要求はドロップされます。ただし、スロットルされたアカウンティング要求は、サーバグループのフェールオーバープロセスによって処理されます。

スロットリング機能は、グローバルまたはサーバグループに対して設定できます。ただし、設定プリファレンスの一般的なルールでは、サーバグループ設定はグローバル設定があれば上書きします。

スロットリング CLI コマンドの構文は次のとおりです。

```
radius-server throttle {[accounting THRESHOLD] [access THRESHOLD [access-timeout
NUMBER_OF-TIMEOUTS]]}
```

値は次のとおりです。

- accounting THRESHOLD：アカウンティング要求のしきい値を指定します。 範囲は 0 ～ 65536 です。 デフォルトは 0 で、スロットリングがアカウンティング要求に対してディセーブルであることを示します。
- access THRESHOLD：アクセス要求のしきい値を指定します。 範囲は 0 ～ 65536 です。 デフォルトは 0 で、スロットリングがアカウンティング要求に対してディセーブルであることを示します。
- access-timeout NUMBER_OF-TIMEOUTS：アクセス要求がドロップされるまでにルータで発生する必要がある連続タイムアウト回数を指定します。 範囲は 0 ～ 10 です。 デフォルトは 3 です。

（注） デフォルトでは、スロットリング機能は BNG でディセーブルです。

スロットリングをグローバルにアクティブ化するには、グローバルな RADIUS スロットリングの設定，（61 ページ）を参照してください。

スロットリングをサーバグループでアクティブ化するには、サーバグループでの RADIUS スロットリングの設定，（63 ページ）を参照してください。

グローバルな RADIUS スロットリングの設定

RADIUS スロットリングをグローバルにアクティブ化するには、次の作業を実行します。

手順の概要

1. **configure**
2. **radius-server throttle access** *threshold_value*
3. **radius-server throttle access** *threshold_value* **access-timeout** *value*
4. **radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. **radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバルコンフィギュレーションモードを開始します。
ステップ 2	**radius-server throttle access** *threshold_value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10`	RADIUS サーバに送信されるアクセス要求の数を制御します。 しきい値は、スロットリングを実行するまでの未処理のアクセス要求数を示します。 範囲は 0 ～ 65535 で、推奨値は 100 です。

	コマンドまたはアクション	目的
ステップ 3	**radius-server throttle access** *threshold_value* **access-timeout** *value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5`	スロットルされたアクセス要求をドロップするまでのタイムアウトの数を指定します。 値は、トランザクションのタイムアウト数を示します。 範囲は 1 ～ 10 で、デフォルトは 3 です。
ステップ 4	**radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5 accounting 10`	RADIUS サーバに送信されるアクセス タイムアウト要求の数を制御します。 しきい値は、スロットリングを実行するまでの未処理のアカウンティング トランザクション数を示します。 範囲は 0 ～ 65535 で、推奨値は 100 です。
ステップ 5	**radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value* 例： `RP/0/RSP0/CPU0:router(config)# radius-server throttle accounting 56 access 10 access-timeout 5`	RADIUS サーバに送信されるアカウンティング要求の数を制御します。 しきい値は、スロットリングを実行するまでの未処理のアカウンティング トランザクション数を示します。 値の範囲は 0 ～ 65535 で、推奨値は 100 です。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

グローバルな RADIUS スロットリングの設定：例

```
configure
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

サーバ グループでの RADIUS スロットリングの設定

サーバ グループで RADIUS スロットリングをアクティブ化するには、次の作業を実行します。

手順の概要

1. **configure**
2. **aaa group server radius** *server_group_name*
3. **server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value*
4. **throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa group server radius** *server_group_name* 例： `RP/0/RSP0/CPU0:router(config)# aaa group server radius SG1`	AAA（RADIUS）サーバグループの定義を設定します。

	コマンドまたはアクション	目的
ステップ 3	**server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# server 99.1.1.10 auth-port 1812 acct-port 1813`	IPアドレスまたはホスト名のいずれか（指定のとおり）でRADIUSサーバのアカウンティング ポートまたは認証ポートを設定します。アカウンティング ポート番号と認証ポート番号の範囲は、0～65535 です。
ステップ 4	**throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value* 例： `RP/0/RSP0/CPU0:router(config-sg-radius)# radius-server throttle access 10 access-timeout 5 accounting 10`	RADIUSサーバに送信されるアクセス要求数とアカウンティング要求数を制御するようにRADIUS スロットリング オプションを設定します。 しきい値は、スロットリングを実行するまでの未処理のアクセス要求数とアカウンティング トランザクション数を示します。 範囲は 0～65535 で、アクセス要求とアカウンティング要求の両方の推奨値は 100 です。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

サーバ グループでの RADIUS スロットリングの設定：例

```
configure
aaa group server radius SG1
server 99.1.1.10 auth-port 1812 acct-port 1813
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

RADIUS の許可変更（CoA）の概要

許可変更（CoA）機能によって、RADIUS サーバはすでに許可されている加入者の許可設定を変更できます。 CoA は、BNG のように、RADIUS サーバから RADIUS クライアントに非同期メッセージを送信できる RADIUS 標準の拡張です。

（注） CoA のサーバは、RADIUS サーバと異なる場合があります。

設定を変更する必要がある加入者を識別するために、RADIUS CoA サーバは、Accounting-Session-ID、Username、IP-Address、および ipv4:vrf-id などのさまざまなキー（RADIUS 属性）をサポートし、使用します。

RADIUS CoA のサポートは、次のとおりです。

- アカウントログイン：ユーザがネットワークにログインするとき、CoA をサポートする外部 Web ポータルが、アカウントログイン要求をユーザ クレデンシャル（ユーザ名およびパスワード）と一緒に BNG に送信します。 BNG のアカウントログインは、これらのクレデンシャルを使用して RADIUS を介してユーザを認証しようとします。
- アカウントログオフ：BNG は加入者の切断イベントとしてアカウントログオフ要求を処理し、セッションを終了します。

  （注） RADIUS CoA サーバは、切断イベントの発生源を区別しません。 したがって、BNG が RADIUS CoA サーバからアカウントログオフ要求を受け取ると、ユーザ開始要求と管理者開始要求の両方で、RADIUS サーバに送信される Acct-Terminate-Cause は常に Admin-Reset として設定されます。

- アカウント更新：BNG は、CoA プロファイルの一部として取得した属性を解析し、適用します。 加入者固有の属性のみが、ユーザ プロファイルでサポートされ、適用されます。

- サービスのアクティブ化：BNGは、加入者に対して事前定義されたサービスを開始します。サービス設定は、動的なテンプレートによってローカルに定義されるか、RADIUS サーバからダウンロードできます。
- サービスの非アクティブ化：BNGは、動的なテンプレートを非アクティブ化することと同様に、加入者に対して以前開始されたサービスを停止します。

CoA からのサービスのアクティブ化

BNG は、CoA 要求によるサービスのアクティブ化をサポートします。 CoA **service-activate** コマンドを使用して、サービスをアクティブ化します。 サービスをアクティブ化するCoA要求には、次の属性が含まれます。

- "subscriber:command=service-activate" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- サービス プロファイルの一部であるその他の属性

重複するサービスのアクティブ化要求を、CoA のサーバから BNG に送信できます。 BNG は、すでにアクティブ化されているサービスのアクションを行いません。 BNGは、次のシナリオの下でCoA サーバに CoA ACK メッセージを送信します。

- 同じパラメータを持つ重複した要求が、すでにアクティブなサービスについてCoAから送信される場合。
- 同じパラメータを持つ重複した要求が、パラメータ化されたサービスを適用するためにCoAから送信される場合。

BNG は、次のシナリオの下で無効な属性としてエラー コードとともに CoA サーバに CoA NACK メッセージを送信します。

- セッションに適用されていないパラメータ化されていないサービスを非アクティブ化するために、要求が CoA から送信される場合。
- セッションに適用されていないパラメータ化されたサービスを非アクティブ化するために、要求が CoA から送信される場合。
- パラメータ化されたサービスを適用する重複リクエストが、CoAからの同一でないパラメータで作成される場合。
- パラメータ化されたサービスを非アクティブ化するために、同一でないパラメータを持つ要求が CoA から送信される場合。

CoA からのサービスの更新

サービス更新機能によって、更新されたサービスを表す新しい RADIUS 属性のリストで既存のサービスプロファイルをアップデートできます。 これは、サービスですでにアクティブ化されている加入者と、今後サービスをアクティブ化する新しい加入者に影響します。 新しい CoA **service-update** コマンドを使用して、この機能をアクティブ化します。 サービスを更新する CoA 要求には、次の属性が含まれます。

- "subscriber:command=service-update" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- サービス プロファイルの一部であるその他の属性

サービス更新 CoA には、少なくとも次の属性が必要です。

- vsa cisco generic 1 string "subscriber:command=service-update"
- vsa cisco generic 1 string "subscriber:service-name=<service name>"

RADIUS ベースの CoA による Web ログイン

Web ログインをサポートするには、一連のポリシー ルール イベントを順番に設定する必要があります。 イベントは、次のとおりです。

- セッションの開始：
  - セッションの開始時に、加入者はインターネット接続を確立するようにセットアップされます。 サービスは、Web ベースのログイン用 Web ポータルに HTTP トラフィックをリダイレクトするためにアクティブ化されます。
  - 認証の待機期間を最大にしてタイマーを開始します。
- アカウントログイン：Web ポータルで、ユーザ名とパスワードなどのユーザ情報が収集され、CoA account-logon コマンドがトリガーされます。 このイベントがトリガーされると、加入者のユーザ名とパスワードが RADIUS サーバによって認証されます。 認証が成功すると、HTTP リダイレクト サービスが非アクティブ化され、すでに接続されているインターネット セットアップへのユーザ アクセスが許可されます。 また、セッションの開始で確立されるタイマーを停止する必要があります。 ただし、アカウントログイン中に認証が失敗した場合、BNG は NAK CoA 要求を送信し、その他の認証の試行を許可します。
- タイマーの期限切れ：タイマーの期限が切れると、加入者セッションは設定に基づいて切断されます。

QoS のサービス アカウンティング

加入者についてイネーブルにされた各サービスのアカウンティング レコードは、設定済みの RADIUS サーバに送信できます。 これらのレコードには、サービスおよび関連カウンタの現在の状態を含む service-start、service-stop、および service-interim レコードを含めることができます。 この機能は、サービス アカウンティング機能です。 サービス アカウンティング レコードは、加入者セッションの一部としてサービスを構成する機能の集まりを表す統合アカウンティング レコードです。

サービス アカウンティングは、サービスがイネーブルになって加入者セッションが有効になると開始されます。 これは、新しいサービスが加入者セッションに適用されるときに、制御ポリシーを介して適用される動的なテンプレートによって、セッションが許可されるときの Access-Accept（AA）メッセージによって、または許可変更（CoA）によって発生することがあります。 サービ

ス アカウンティングは、セッションが終了するとき、またはサービスを非アクティブ化する一部の他のイベントやCoAによってサービスがセッションから削除されるときに停止します。開始レコードにカウンタはありません。QoS カウンタを持つ中間レコードと停止レコードは、サービスアカウンティングが QoS に対してイネーブルになると生成されます。 中間アカウンティング レコードは、事前定義された定期的なオプションとして、開始または停止アカウンティングの中間で生成できます。 中間期間がゼロの場合、中間アカウンティング レコードは作成されません。異なる中間インターバルは、セッションごとにすべてのサービスに基づきます。サービスアカウンティングは、設定に基づいて各テンプレートでイネーブルになります。

（注） 動的なテンプレートに関連付けられたポリシーマップを編集して、サービス パラメータを変更できます。 ただし、これによってアカウンティング レコードは更新されません。 したがって、すべてのアカウンティング レコードを正確に生成するには、CoA を介して、すべての必要なサービス パラメータを持つ新しいサービスを作成し、新しいサービスに関連付けることを推奨します。

サービス アカウンティングでは、特定の加入者に対して各サービスの下に適用される入力および出力 QoS ポリシーの統計情報を、アカウンティングの中間レコードと停止レコードの一部として報告する必要がある場合があります。 各サービスでは、次の QoS カウンタをアカウンティング レコードの一部として報告できます。

- BytesIn：ポリサー ドロップを引いたサービスの入力 QoS ポリシーのすべてのクラスに一致するバイトの集合体です。
- PacketsIn：ポリサー ドロップを引いたサービスの入力 QoS ポリシーのすべてのクラスに一致するパケットの集合体です。
- BytesOut：キューイング ドロップを引いたサービスの出力 QoS ポリシーのすべてのクラスに一致するバイトの集合体です。
- PacketsOut：キューイング ドロップを引いたサービスの出力 QoS ポリシーのすべてのクラスに一致するパケットの集合体です。

アカウンティング統計情報の収集をサポートし、統計情報を AAA サービス アカウンティング レコードで報告する必要がある動的なテンプレート機能は、新しく導入された任意の **acct-stats** 設定オプションを使用して、それらの機能でアカウンティング統計情報をイネーブルにできます。 このオプションは、統計情報の収集をサポートしない機能では使用できません。 デフォルトでは、QoS アカウンティング統計情報は、パフォーマンスを最適化できません。

（注） 各方向の QoS カウンタは、QoS ポリシーが特定の方向でそのサービスに適用される場合に限り報告されます。 たとえば、サービスに適用される入力ポリシーがない場合、BytesIn および PacketsIn カウンタは 0 として報告されます。

前提条件

- 加入者アカウンティング、サービス アカウンティングの親アカウンティング レコードは、サービス アカウンティング機能が機能できるように設定される必要があります。
- キーワード **acct-stats** は、サービスポリシー コンフィギュレーションで、サービス アカウンティング機能がレコードの一部として機能カウンタ情報を報告できるように設定される必要があります。

制約事項

- IPv4 および IPv6 の加入者セッションには、単一のサービス アカウンティング レコードのセットがあります。 これらは、1 セットの bytes_in、bytes_out、packets_in、packets_out カウンタにマージされます。

サービス アカウンティングの設定

動的なテンプレートを介してサービス アカウンティングを設定するには、次の作業を実行します。

はじめる前に

次の作業を実行する前に、加入者アカウンティングを設定する必要があります。 設定手順については、IPv4 または IPv6 加入者セッションの動的なテンプレートの作成, （96 ページ）を参照してください。

手順の概要

1. **configure**
2. **aaa accounting service** {*list_name* | **default**} {**broadcast group** {*group_name* | **radius**} |**group** {*group_name* | **radius**} }
3. **aaa service-accounting** [**extended** | **brief**]
4. **dynamic-template**
5. **type service** *dynamic-template-name*
6. **accounting aaa list** {*method_list_name* | **default**} **type service** [**periodic-interval** *time*]
7. {**ipv4** | **ipv6**} **access-group** *access-list-name*
8. **service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**]
9. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa accounting service** {*list_name* | **default**} {**broadcast group** {*group_name* | **radius**} |**group** {*group_name* | **radius**} } 例： `RP/0/RSP0/CPU0:router(config)# aaa accounting service l1 group srGroup1`	サービス アカウンティングのアカウンティング リストを作成します。
ステップ 3	**aaa service-accounting** [**extended** | **brief**] 例： `RP/0/RSP0/CPU0:router(config)# aaa service-accounting brief`	（任意）加入者アカウンティングの状態のレベルを選択するため、および簡潔な形式または拡張形式で報告する属性を特定するために、サービスのアカウンティング パラメータを設定します。 （注） デフォルト設定では、拡張です。
ステップ 4	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 5	**type service** *dynamic-template-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service s1`	サービスのユーザ定義名で動的なテンプレートを作成します。
ステップ 6	**accounting aaa list** {*method_list_name* | **default**} **type service** [**periodic-interval** *time*] 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# accounting aaa list l1 type service periodic-interval 1000`	サービス アカウンティング機能を設定します。

	コマンドまたはアクション	目的

ステップ 7	{**ipv4** | **ipv6**} **access-group** *access-list-name*
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 access-group ACL1`
	
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 access-group ACL2`
		インターフェイスに IPv4 または IPv6 アクセス リストを設定します。

ステップ 8	**service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input QoS1 acct-stats`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy output QoS2 acct-stats`
		サービス ポリシーを動的なテンプレートに関連付け、**acct-stats** キーワードを使用してサービス アカウンティング機能をイネーブルにします。

ステップ 9	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

サービス アカウンティングの設定：例

```
configure
 aaa accounting service S1 group SG1
 aaa service-accounting brief
 dynamic-template
  type service s1
   accounting aaa list S1 type service periodic-interval 600
   ipv4 access-group ACL1
   service-policy input QOS1 acct-stats
   service-policy output QOS2 acct-stats
!
!
end
```

統計情報インフラストラクチャ

アカウンティング カウンタは、サービス アカウンティング統計情報 ID（statsD）インフラストラクチャによって維持されます。サービスアカウンティングは、次の方法で統計情報インフラストラクチャと対話します。

- 各機能には、その機能の統計情報カウンタを戻す統計情報収集機能プロセスがあります。
- 単一の収集機能が、複数の機能のカウンタを処理できます。
- アカウンティング プロセス、サービス アカウンティング管理エージェントは、通知の登録と統計情報の要求にアクセス ライブラリを使用し、RADIUS サーバにプッシュします。

statsD からデータをプルするポーリング期間があります。 停止レコードでサブセカンド精度をサポートするには、セッションが終了したら、正確なデータを得るためにポーリング方法を待つことなく、ただちに統計情報をプルします。 セッション アカウンティングおよびサービス アカウンティングで同じ方法を続けます。サブセカンド精度は、中間アカウンティング レコードの送信中にプルされるデータがないため、中間レコードで報告されるデータではサポートされません。

統計情報 ID（statsD）の設定

statsD は、デフォルトでは 900 秒ごと（15 分ごと）に機能の統計情報をポーリングするように設定されています。 デフォルトの数値を変更し、ポーリング間隔を増減するには、次の作業を実行します。

手順の概要

1. **configure**
2. **statistics period service-accounting** {*period* | **disable**}
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**statistics period service-accounting** {*period*	**disable**} 例： `RP/0/RSP0/CPU0:router(config)# statistics period service-accounting 1800`	サービス アカウンティング機能の統計情報収集機能の収集期間を設定します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

サービス アカウンティングの設定：例

```
configure
 statistics period service-accounting 1800
end
```

Per-VRF AAA 機能について

Per VRF AAA 機能によって、仮想ルーティング/転送（VRF）インスタンスに基づいた、認証、許可、アカウンティング（AAA）を実行できます。 この機能によって、プロバイダーエッジ（PE）または仮想ホーム ゲートウェイ（VHG）で、カスタマーのバーチャル プライベート ネットワーク（VPN）に関連付けられたカスタマーの RADIUS サーバと、RADIUS プロキシを経由せずに直接通信できます。

ISP は、AAA サーバ グループ、方式リスト、システム アカウンティング、およびプロトコル固有のパラメータなどの動作パラメータを定義し、特定の VRF インスタンスにこれらのパラメータを関連付ける必要があります。

Per VRF AAA 機能は、サーバグループ、RADIUS、およびシステム アカウンティング コマンドに対する VRF の拡張でサポートされます。 サーバ グループ内のサーバのリストは、グローバル コンフィギュレーションでのホストへの参照に加えて、プライベート サーバの定義を含むように拡張されます。 これによって、カスタマー サーバとグローバル サービス プロバイダーのサーバの両方に同時にアクセスできます。 Per-VRF AAA をグローバルに設定するために使用するコマンドの構文は、次のとおりです。

```
radius source-interface subinterface-name [vrf vrf-name]
```

RADIUS ダブルディップ機能

BNG は、BNG が初期認証または許可要求をサービス プロバイダーの RADIUS サーバに送信する場合に、加入者セッションに関連付けられた正しい VRF に順番に応答する RADIUS ダブルディップ機能をサポートします。 その後、BNG は、元の要求をリダイレクトし、指定された VRF に関連付けられた適切な RADIUS サーバに、2 番目の要求として送信します。

その他の関連資料

ここでは、RADIUS の実装に関連する参考資料を示します。

RFC

標準/RFC - AAA	
RFC-2865	『Remote Authentication Dial In User Service (RADIUS)』
RFC-2866	『RADIUS Accounting』
RFC-2867	『RADIUS Accounting Modifications for Tunnel Protocol Support』
RFC-2868	『RADIUS Attributes for Tunnel Protocol Support』
RFC-2869	『RADIUS Extensions』
RFC-3575	『IANA Considerations for RADIUS』
RFC-4679	『DSL Forum Vendor-Specific RADIUS Attributes』
RFC-5176	『Dynamic Authorization Extensions to RADIUS』

MIB

MIB	MIB のリンク
—	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/cisco/web/support/index.html

第 4 章

コントロール ポリシーのアクティブ化

コントロール ポリシーによって、サービス プロバイダーは加入者のさまざまなライフサイクル イベント中に実行される特定のアクションを定義できます。 この章では、BNG ルータでのコントロール ポリシーのアクティブ化に関する情報を提供します。 コントロール ポリシーは、ポリシーマップを使用して定義されます。 ポリシーマップには、一連のイベント、つまり特定のアクションの実行中のイベントが含まれます。 アクションを実行するための条件は、クラスマップで定義されます。 クラスマップは作成された後、ポリシーマップに含まれます。 ポリシーマップは、ポリシーのルータ インターフェイス上でアクティブ化され、有効になります。 ポリシーマップによって実行できるアクションの1つは、動的なテンプレートをアクティブ化することです。 動的なテンプレートは、加入者のグループに適用される一連の設定項目をグループ化するために使用されるコンテナです。 この章は、次の内容で構成されています。

コントロール ポリシーの概要

コントロール ポリシーによって、サービス プロバイダーは、セッションの作成や接続の切断など、加入者のさまざまなライフサイクル イベント中に実行される必要のあるアクションを定義できます。 イベントの完全なリストについては、コントロール ポリシー イベント，（81 ページ）を参照してください。

さまざまな一致基準に基づいて、異なるユーザに対して異なるアクションを実行できます。 コントロール ポリシーで指定できる一部のアクションは、次のとおりです。

- 外部 AAA サーバによる加入者の認証または許可

- 加入者アカウンティングの開始
- 動的なテンプレートを使用した加入者に固有の設定のアクティブ化

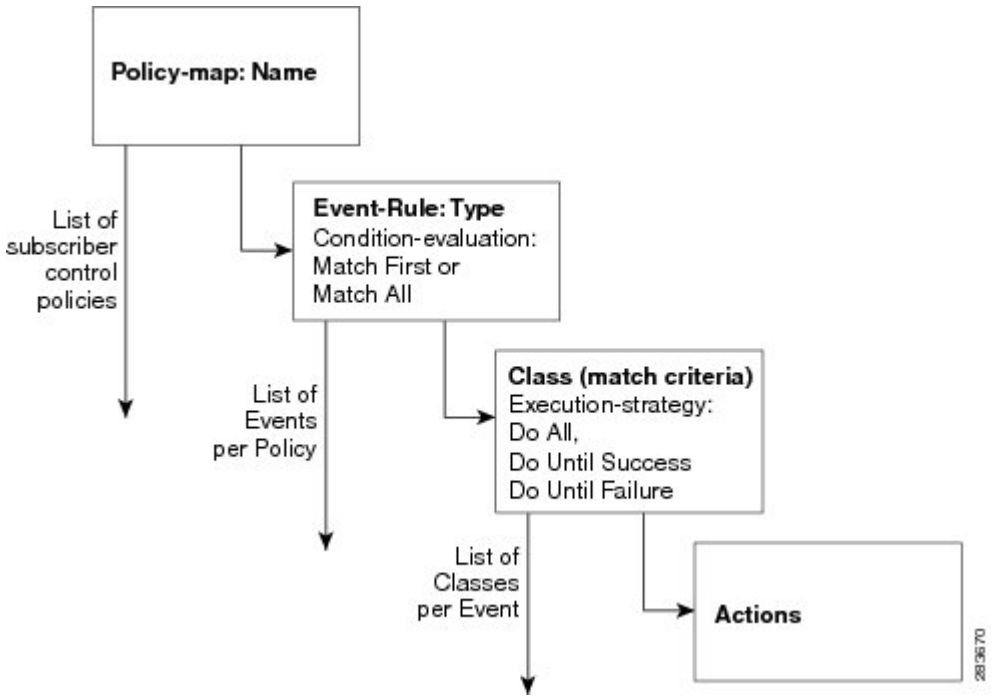
コントロールポリシーは、ポリシーマップおよびクラスマップを使用して導入されます。各ポリシーマップには、サービス プロバイダーが加入者のライフサイクルに適用できると見なしたイベントのリストが含まれます。 ポリシーマップは、これらのイベント中に実行されるアクションも定義します。ただし、これらのアクションは、特定の条件を満たす場合にのみ実行されます。これらの条件は、一致基準と呼ばれます。 一致基準は、ポリシーマップ内に含まれるクラスマップで定義されます。 異なる加入者に対して異なる一致基準を持つことができます。

たとえば、特定の「MAC アドレス」一致基準の「セッションの開始」イベントが発生したときに、「加入者の認証」アクションを開始するようにコントロールポリシーを作成できます。 このコントロールポリシーを導入した後、指定されたMACアドレスを持つデバイスが新しいセッションを開始すると、BNG は加入者認証プロセスを開始します。

ポリシーマップで定義されているアクションは、アクションハンドラによって実行されます。サポートされるアクション ハンドラの詳細については、アクション ハンドラ, （383 ページ）を参照してください。

次の図は、コントロールポリシーの構造を示しています。図は、各ポリシーに複数のイベントがあること、各イベントに複数のクラスがあること、各クラスに複数のアクションがあることを示しています。 結果として、1 つまたは多数のイベント中に、単一または複数の条件について一致が検出されると、単一のポリシー マップを使用して複数のアクションをトリガーできます。

図 *4*：コントロール ポリシー



次の設定例は、コントロール ポリシー構造を示しています。

```
policy-map type control subscriber policy-map-name
   event <event-type> [match-all|match-first]
      class type control subscriber <class-map-name>
           <seq#> <action-type> <action_options>
```

クラスマップの作成

クラスマップを使用して、トラフィック クラスを定義します。トラフィックは、クラスマップで定義されている一致基準に基づいて分類されます。一致基準のパラメータは、プロトコル、MAC アドレス、入力インターフェイス、アクセス グループなどになります。

複数の一致基準が 1 つのクラスマップに表示される場合、一致基準をどのように評価するかの定義を手順に含める必要があります。 評価手順には、次の 2 つのタイプがあります。

- Match-any：評価されるトラフィックが指定された基準のいずれかと一致する場合に、明確に一致します。
- Match-all：評価されるトラフィックが指定されたすべての条件と一致する場合に、明確に一致します。

一致すると、トラフィックはクラスのメンバーと見なされます。

各クラスマップは、ID に名前を割り当てられます。クラスマップ名は、ポリシーマップ内で指定されます。

クラスマップの作成については、クラスマップの設定, （79 ページ）を参照してください。

クラスマップの設定

コントロールポリシーのクラスマップを設定するには、次の作業を実行します。たとえば、このクラスマップは、評価手順「match-any」で作成されます。 一致基準は、値「PPP」の「プロトコル」です。 結果として、セッションが PPP プロトコルを使用する場合、明確に一致します。

手順の概要

1. **configure**
2. **class-map type control subscriber match-any** *class-map-name*
3. **match protocol ppp**
4. **end-class-map**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**class-map type control subscriber match-any** *class-map-name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-any clmap1`	ユーザ定義名を使用して新しい加入者のコントロール クラスマップを作成します。 クラスマップ モードを開始します。 一致評価手順を「match-any」として定義します。
ステップ 3	**match protocol ppp** 例： `RP/0/RSP0/CPU0:router(config-cmap)# match protocol ppp`	一致基準を PPP プロトコルとして定義します。 （注） 複数の match ステートメントは、クラスマップごとに適用できます。
ステップ 4	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	クラス マップの設定を終了します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

クラスマップの設定：例

```
class-map type control subscriber match-any DHCP_class
match protocol dhcpv4
end-class-map
 !
!
end
```

ポリシーマップの作成

ポリシーマップは、クラスマップの定義に基づいて一致する場合、一連のアクションが実行されるイベントを定義するために使用されます。 サポートされる BNG イベントの詳細については、コントロール ポリシー イベント, （81 ページ）を参照してください。

ポリシーマップは、一連のイベントを示します。 イベントごとに、一連のクラスマップを定義します。 各クラスマップでは、一連のアクションが順番に表示されます。 ポリシーマップが BNG ルータ インターフェイス上で適用されると、トラフィックがクラスマップに記載されている基準を満たすときに、アクションが実行されます。

複数のクラスマップがポリシーマップに表示されている場合、手順を指定して、どのクラスマップを適用するのかを定義する必要があります。 評価手順には、次の 2 つのタイプがあります。

- First-match：最初のクラスマップに対して一致した場合にのみ、アクションが実行されます。
- Match-all：すべての一致するクラスに対して、アクションが実行されます。

クラスマップと同様に、各ポリシーマップは ID に名前を割り当てられます。 ポリシーマップ名は、ルータ インターフェイスでポリシーマップをアクティブ化するときに指定されます。

ポリシーマップの作成については、ポリシーマップの設定, （83 ページ）を参照してください。

コントロール ポリシー イベント

BNG のコントロール ポリシーは、次に示すイベントをサポートします。 次のイベントは、タスクポリシーマップの設定, （83 ページ）を使用するポリシーマップの作成中に定義される必要があります。

- セッションの開始：このイベントは、PPPoE と DHCP アクセス プロトコルによって使用され、ポリシー プレーンで加入者を作成します。 オペレータは、AAA アクションを設定し、加入者に適した動的なテンプレートをアクティブ化することがあります。
- セッションのアクティブ化：一部のアクセス プロトコルには、2 段階のセッション起動が必要です。たとえば、PPPoE 加入者によって、PPPoE アクセス プロトコルは、first sign of life（FSOL）のセッションの開始イベントをコールします。この後、PPP ネゴシエーションおよび認証中にセッションのアクティブ化が続きます。 オペレータは、AAA アクションを設定し、加入者に適しているとして動的なテンプレートをアクティブ化します。
- サービスの停止：CoA は、このイベントを生成する責任があります。 BNG のオペレータは、アクションのアクティブ化または非アクティブ化を設定し、サービスが停止されると加入者をデフォルトの状態にします。
- 認証応答なし：設定されている場合、AAA サーバから認証要求の応答がない場合にこのイベントがトリガーされます。 このイベントによって、ネットワーク アクセス サーバ（NAS）のオペレータは障害の処理方法を定義できます。 認証応答なしイベントが設定されていない場合、認証失敗の結果は、デフォルトの処理用のアクセス プロトコルに伝播されます。
- 許可応答なし：設定されている場合、AAA サーバから許可要求の応答がない場合にこのイベントがトリガーされます。 このイベントによって、NAS オペレータは障害の処理方法を定義できます。 許可応答なしイベントが設定されていない場合、許可結果はデフォルトの処理用のアクセスプロトコルに伝播され、これによって許可をトリガーしたクライアントが加入者セッションを切断します。
- 認証失敗：設定されている場合、および RADIUS サーバが認証失敗を返す場合、ポリシー ルール エンジンは、加入者を切断しないように要求を出したクライアントに「認証成功」を返します。 さらに、必要な動作を提供するクライアントに依存するのではなく、設定された認証失敗イベント内のアクションを加入者に適用します。
- 許可失敗：許可の失敗イベントは、RADIUS サーバのアクセス要求の拒否を示します。 設定されている場合、サービス プロバイダーはクライアントからの失敗のデフォルトの処理を上書きします。
- 時間が決まったポリシーの期限切れ：設定されている場合、このイベントは、加入者セッションで設定されているポリシーのタイマー設定アクションの結果としてトリガーされます。 このイベントによって、NAS オペレータは多数のシナリオのタイマーを定義できます。 セット タイマーは、特定の加入者の状態の変更が行われたことを示します。 セッションが目的の状態にない場合、NAS オペレータは設定された切断アクションによってセッションを切断または終了したり、異なるユーザ ポリシーを課すことができます。
- アカウントログイン：設定されている場合、このイベントは、デフォルトのアカウントログイン処理に動作を上書きします。 デフォルトの動作は、提供されたクレデンシャルを使用した認証をトリガーするだけです。 ただし、デフォルトのアカウントログイン イベントを上書きする場合、明示的に、認証アクションと必要なその他のアクションを設定する必要があります。
- アカウントログオフ：設定されている場合、このイベントは、デフォルトのアカウントログオフ処理の動作を上書きします。 アカウントログオフ処理のデフォルトの動作は、加入者を切断することです。 デフォルトの動作を上書きできることは有用です。 加入者を切断する

のではなく、サービス プロバイダーは再認証を実行できます。 加入者に対して HTTP リダイレクト機能をイネーブルにすることによって、新しいアカウントログインで再認証が実行されます。

ポリシーマップの設定

コントロール ポリシーのポリシー マップを設定するには、次の作業を実行します。 たとえば、このポリシーマップは、セッションの開始およびセッションのアクティブ化イベントに対して作成されます。 セッションの開始イベントでは、動的なテンプレートがアクティブ化されます。 セッションのアクティブ化イベントでは、認証プロセスが起動されます。 サポートされるイベントの詳細については、コントロール ポリシー イベント, （81 ページ）を参照してください。

手順の概要

1. **configure**
2. **policy-map type control subscriber** *policy-map-name*
3. **event session-start match-all**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*
6. **event session-activate match-all**
7. **class type control subscriber** *class_name* **do-until-failure**
8. *sequence_number* **authenticate aaa list default**
9. **end-policy-map**
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type control subscriber** *policy-map-name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber plmap1`	ユーザ定義名で新しいポリシーマップを作成します。 ポリシーマップ モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**event session-start match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	アクションを実行するイベント（セッションの開始）を定義します。 すべての一致したクラスにアクションを実行する「match-all」として一致手順を定義します。
ステップ 4	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber CL1 do-until-failure`	イベントにクラスマップを関連付けます。 クラスマップ名を指定する必要があります。 障害が発生するまでアクションを実行するように指定します。
ステップ 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template template1`	実行されるアクションを定義します。 この場合、動的なテンプレートがアクティブ化されます。 （注） このコマンドを繰り返して、複数のアクションを定義できます。
ステップ 6	**event session-activate match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-activate match-all`	アクションを実行するイベント（セッションのアクティブ化）を定義します。 すべての一致したクラスにアクションを実行する「match-all」として一致手順を定義します。
ステップ 7	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber CL1 do-until-failure`	イベントにクラスマップを関連付けます。 クラスマップ名を指定する必要があります。 障害が発生するまでアクションを実行するように指定します。
ステップ 8	*sequence_number* **authenticate aaa list default** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 2 authenticate aaa list default`	実行されるアクションを定義します。 この場合、AAA リストの認証を開始します。
ステップ 9	**end-policy-map** 例： `RP/0/RSP0/CPU0:router(config-pmap)# end-policy-map`	ポリシー マップの設定を終了します。
ステップ 10	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ポリシーマップの設定：例

```
policy-map type control subscriber PL1
 event session-start match-first
  class type control subscriber DHCP_class do-until-failure
   1 activate dynamic-template dhcp
  class type control subscriber class-default do-until-failure
! Packet trigger is default
   1 activate dynamic-template packet-trigger
 end-policy-map
 !
!
end
```

ポリシーマップのアクティブ化

ポリシーマップを作成した後、ルータ インターフェイスでアクティブ化する必要があります。ポリシーは、ポリシ マップのアクティブ化が完了した後にのみ実行されます。 1 つまたは複数のポ

リシーマップは、サービスポリシーを設定します。サービスポリシーをイネーブルにするには、加入者インターフェイスでのサービスポリシーのイネーブル化, （86 ページ）を参照してください。

加入者インターフェイスでのサービスポリシーのイネーブル化

加入者インターフェイスでサービスポリシーをイネーブルにするには、次の作業を実行します。プロセスには、以前に作成したポリシーマップのインターフェイスへの接続が含まれます。 このプロセスを完了すると、クラスマップで定義されたアクションが、サービスポリシーがイネーブルなインターフェイスに着信するトラフィックで有効になります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type control subscriber** *policy_name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.10`	Bundle-Ether アクセス インターフェイスのインターフェイス コンフィギュレーション モードを開始します。 （注） IPoE セッションでは、**arp learning disable** コマンドを使用して、ダイナミック ARP ラーニングをアクセスインターフェイスでディセーブルにすることを推奨します。
ステップ 3	**service-policy type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber plmap1`	「plmap1」という名前の事前定義されたポリシーマップをアクセス インターフェイスに適用します。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

動的なテンプレートの定義

動的なテンプレートは、一連の設定をグループ化し、加入者セッションに適用するために使用されるコンテナです。 動的なテンプレートは、CLI でグローバルに設定されます。 ただし、動的なテンプレートを定義することによって、設定が加入者インターフェイスにすぐに適用されるわけではありません。 動的なテンプレート内の設定は、動的なテンプレートがコントロールポリシーを使用してアクティブ化される場合にのみ、加入者インターフェイスに適用されます。 同様に、適用された設定は、動的なテンプレートがコントロール ポリシーを使用して非アクティブ化される場合にのみ、停止します。

動的なテンプレートには、次の 3 つの基本タイプがあります。

- PPP テンプレート：PPPoE プロトコルに関連する特定の設定が含まれます。
- IP 加入者テンプレート：IP 加入者セッションでアクティブ化される特定の設定が含まれます。

- サービス テンプレート：セッション ライフサイクル イベントに応じてアクティブ化されるサービス関連の設定が含まれます。 サービス テンプレートは、インターフェイスまたはメディア固有コマンドを含みません。

動的なテンプレートは、CLI で設定するか、AAA サーバからダウンロードすることができます。次の設定例では、ポリシー マップによって、CLI で定義された IP 加入者の動的なテンプレートがアクティブ化されます。

```
dynamic-template
type ipsubscriber ipsub
ipv4 unnumbered Loopback400
policy-map type control subscriber PL2
event session-start match-first
class type control subscriber class-default do-all
1 activate dynamic-template ipsub
```

AAA サーバからダウンロードされる動的なテンプレートには、ユーザ プロファイルとサービス プロファイルの 2 つのタイプがあります。 ユーザ プロファイルは、単一の加入者に適用され、サービス プロファイルは、複数の加入者に適用できます。 次の設定例では、ポリシー マップは、AAA サーバからサービス テンプレートをダウンロードします。

```
Radius Config:
service1 Password="xxxxxx"
Cisco-avpair = "ipv4:ipv4-unnumbered=Loopback400"

Router Config:
policy-map type control subscriber PL2
event session-start match-first
class type control subscriber class-default do-all
1 activate dynamic-template service1 aaa list default
```

前述の例では、「aaa list default」キーワードが、テンプレート「service1」を AAA サーバからダウンロードするよう指定しています。テンプレートは、一度だけダウンロードされます。 service1 を示す複数のコントロール ポリシーがある場合は、前にダウンロードしたバージョンを取得します。

同じイベントまたは異なるイベントに対して、同じ加入者インターフェイス上で複数の動的なテンプレートをアクティブ化できます。 特定の機能の設定が複数の動的なテンプレートで定義されている場合、設定は、特定の優先順位ですべてのテンプレートから取得されます。 この順位は、動的なテンプレートのタイプ、および CLI または AAA のいずれから適用されているかに基づきます。 順序は次のようになります。

- AAA からユーザ プロファイルによって適用されたテンプレート
- AAA からサービス プロファイルによって適用されたテンプレート
- CLI から適用された IP 加入者テンプレート
- CLI から適用された PPP テンプレート
- CLI から適用されたサービス テンプレート

動的なテンプレートを使用して特定の機能設定を定義するタスクには、対応する機能のトピックが含まれます。

その他の関連資料

ここでは、コントロール ポリシーの実装に関連する参考資料を示します。

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

第 5 章

加入者セッションの確立

加入者は、加入者セッションと呼ばれる論理接続経由でネットワーク リソースにアクセスします。 この章では、加入者セッション、つまり IPoE および PPPoE のさまざまなタイプ、および DHCP による IP アドレッシングについて説明します。 この章の内容は、次のとおりです。

加入者セッションの概要

セッションは、顧客宅内装置（CPE）とネットワーク リソース間の論理接続を表します。 ネットワーク リソースへの加入者のアクセスをイネーブルにするには、ネットワークは、加入者とのセッションを確立する必要があります。 各セッションの確立は、次のフェーズで構成されます。

- 接続の確立：このフェーズでは、CPE が通信する BNG を検出します。
- 加入者の認証および許可：このフェーズでは、BNG が加入者を認証し、ネットワークの使用を許可します。 このフェーズは、RADIUS サーバを使用して行われます。
- 加入者への ID の付与：このフェーズでは、加入者は ID と IP アドレスを割り当てられます。
- セッションのモニタリング：このフェーズでは、BNG がセッションが稼働中であることを確認します。

加入者は、BNG で直接設定されません。 代わりに、加入者の機能と加入者セッションが動的に起動および停止するフレームワークが作成されます。 フレームワークは、次の機能を実行するコントロール ポリシーと動的なテンプレートで構成されます。

- コントロールポリシーは、セッションの開始要求の受信などの特定のイベント、または認証の失敗が発生したときに、BNG が実行するアクションを決定します。 アクションは、コントロール ポリシーで定義されたクラスマップで決定されます。 アクションには、動的なテンプレートのアクティブ化が含まれます。
- 動的なテンプレートには、加入者セッションに適用される一連の CLI コマンドが含まれています。 複数の動的なテンプレートは、同じ加入者インターフェイス上で、一度にアクティブ化できます。 また、異なるコントロール ポリシーを介して複数の加入者インターフェイスで、同じ動的なテンプレートをアクティブ化することもできます。

サービス プロバイダーは、次の方法で VLAN 経由で加入者を導入できます。

- 1:1 VLAN モデル：このモデルは、1 つの専用 VLAN を各カスタマーに使用できる場合のシナリオを示します。 各 VLAN は、内部 VLAN タグが加入者を表し、外部 VLAN タグが DSLAM を表す場合の q-in-q VLAN です。
- N: 1 VLAN モデル：このモデルは、複数の加入者を共有 VLAN で使用できる場合のシナリオを示します。 VLAN タグは、DSLAM または集約デバイスを表します。
- あいまいな VLAN：このモデルによって、オペレータは単一 CLI 行で多数の VLAN を指定できます。 あいまいな VLAN を使用して、内部または外部タグ（または両方）の範囲を VLAN サブインターフェイス上で設定できます。 これは、すべての加入者が VLAN タグ セットに一意の値を持つ 1:1 モデルで特に便利です。 あいまいな VLAN の詳細については、あいまいな VLAN の加入者セッション，（293 ページ）を参照してください。

加入者セッションは、仮想インターフェイスである加入者インターフェイスを介して確立されます。 各加入者セッションに 1 つのみインターフェイスを作成できます。 ポートには複数の VLAN を含めることができ、それぞれが複数の加入者をサポートできます。 BNG は、各種のセッションの加入者インターフェイスを作成します。 これらのインターフェイスは、Bundle-Ether2.100.pppoe312 などの親インターフェイスに基づいて命名されます。 バンドル（またはバンドル VLAN）インターフェイス上の加入者は冗長性を確保し、BNG のルート プロセッサ（RP）で管理されます。

ネットワークの冗長性とロード バランシングを提供するために、サービス プロバイダーは、DSLAM と BNG 間に複数のリンクを導入できます。 個々のリンクは、Ether-Bundle 経由の VLAN またはリンク集約グループ（LAG）を含む Ether-Bundle にグループ化できます。 加入者セッションは、バンドルまたはグループ内の任意のリンクでアクティブ化できます。 BNG が LAG 設定で導入されている場合、1 つの加入者に対するすべてのトラフィックを、Ether-Bundle の 1 つのリンクを経由するように設定する必要があります。 ロードバランシングは、異なるリンクにさまざまな加入者を配置することによって実現されます。

加入者セッションを確立する 2 種類のメカニズム（IPoE および PPPoE）があります。 これらは、次のトピックで説明します。

（注） すべての加入者セッションをクリアするために **clear subscriber session all** コマンドが発行される場合、およびセッション停止の進行中にルートプロセッサフェールオーバー（RPFO）が発生した場合、RPFO 後に同じコマンドを再実行して残りのセッションがある場合はそのすべてが停止されるようにすることを推奨します。

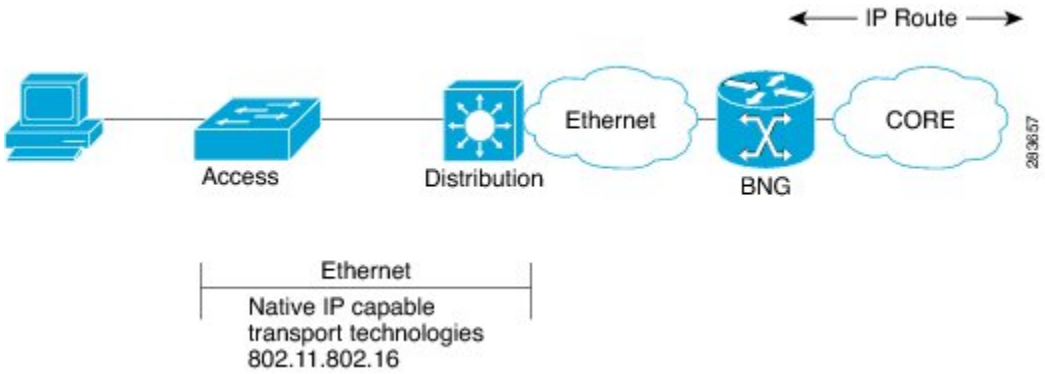
制約事項

- BNG では、加入者の動的な作成のみがサポートされます。 また、加入者は、バンドル インターフェイスにのみ存在する必要もあります。
- 加入者の VRF がアクセス インターフェイスの VRF 値から取得されている場合、加入者が使用する動的なテンプレートに設定された VRF は一致する必要があります。 2 つの VRF が一致しない場合、セッションは正常に終了していません。

IPoE セッションの確立

IPoE 加入者セッションでは、加入者が CPE デバイスで IPv4 または IPv6 を実行し、レイヤ 2 集約ネットワークまたはレイヤ 3 ルーテッド ネットワーク経由で BNG に接続します。 レイヤ 2 集約ネットワーク経由で接続する IP 加入者セッションは、L2 接続と呼ばれます。 IPoE 加入者セッションは、常に BNG で終了し、サービス プロバイダー ネットワークにルーティングされます。 IPoE は、DHCP に基づいて IP アドレスを割り当てます。 一般的な IPoE セッションを次の図に示します。

図 *5*：*IPoE* セッション



IPoE セッションのプロビジョニング プロセスは、次のとおりです。

- アクセス インターフェイスでの IPv4 または IPv6 プロトコル処理のイネーブル化。 アクセス インターフェイスでの IPv4 または IPv6 のイネーブル化, （94 ページ）を参照してください。

（注） 加入者の導入については、アクセスインターフェイス コンフィギュレーション モードで **arp learning disable** コマンドを使用して、アクセスインターフェイスでダイナミック ARP ラーニングをディセーブルにすることを推奨します。

- IPoE セッションの設定を含む動的なテンプレートの作成。 IPv4 または IPv6 加入者セッションの動的なテンプレートの作成，（96 ページ）を参照してください。
- 動的なテンプレートをアクティブ化するポリシーマップの作成。 IPoE セッション中に実行されるポリシーマップの作成，（99 ページ）を参照してください。
- サービスポリシーのアクティブ化によるアクセス インターフェイスでの IPoE 加入者作成のイネーブル化。 サービスポリシーは、アクセス インターフェイスにポリシーマップを適用します。 アクセス インターフェイスでの IPoE 加入者のイネーブル化，（102 ページ）を参照してください。

アクセス インターフェイスでの IPv4 または IPv6 のイネーブル化

アクセス インターフェイスでの IPv4 および IPv6 処理をイネーブルにするには、次の作業を実行します。 この例では、IPv4 は、アンナンバードバンドルインターフェイスでプロビジョニングされています。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **arp learning disable**
4. **ipv4 unnumbered** *interface-type interface-instance*
5. **ipv6 enable**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.10`	バンドルインターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**arp learning disable** 例： `RP/0/RSP0/CPU0:router(config-if)# arp learning disable`	アクセスインターフェイスの ARP ラーニングをディセーブルにします。
ステップ 4	**ipv4 unnumbered** *interface-type interface-instance* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5`	明示的な IPv4 アドレスをインターフェイスに割り当てることなく、アンナンバード インターフェイス上での IPv4 処理をイネーブルにします。 代わりに、IP アドレスは、ループバック インターフェイスから流用されます。「ipv4 unnumbered」コマンドの場合、IP アドレスが割り当てられ、**show interfaces** コマンドに対して状態が「UP」と表示される、同じルータの別のインターフェイスを指定する必要があります。
ステップ 5	**ipv6 enable** 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	明示的な IPv6 アドレスが割り当てられていないアンナンバード インターフェイスでの IPv6 処理をイネーブルにします。 （注） この手順は、インターフェイスでの IPv6 処理をイネーブルにするだけでなく、それに対する IPv6 リンクローカル ユニキャスト アドレスを割り当てます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アクセス インターフェイスでの IPv4 または IPv6 のイネーブル化：例

```
//Enabling IPv4 on an Access Interface

configure
interface Bundle-Ether100.10
arp learning disable
ipv4 unnumbered loopback 5
!
!
end


//Enabling IPv6 on an Access Interface

configure
interface Bundle-Ether100.10
arp learning disable
ipv6 enable
!
!
end
```

IPv4 または IPv6 加入者セッションの動的なテンプレートの作成

IPv4 または IPv6 加入者セッションの動的なテンプレートを作成するには、次の作業を実行します。 たとえば、この動的なテンプレートでは、IPv4 または IPv6 セッションの MTU 値を指定し、uRPF をイネーブルにします。 uRPF は、変造または偽造された IPv4 送信元アドレスからのトラフィックを加入者インターフェイスで受け入れられないようにします。 uRPF 機能の詳細については、uRPF, （296 ページ）を参照してください。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type** { **ipsubscriber** |**ppp** |**service** } *dynamic-template-name*
4. **timeout idle** *value*
5. **accounting aaa list default type session periodic-interval** *value* **dual-stack-delay** *value*
6. {**ipv4** |**ipv6**} **mtu** *mtu-bytes*
7. {**ipv4** |**ipv6**}**verify unicast source reachable-via rx**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 3	**type** { **ipsubscriber** |**ppp** |**service** } *dynamic-template-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipsub1`	IP 加入者のユーザ定義名を使用して動的なテンプレートを作成します。
ステップ 4	**timeout idle** *value* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# timeout idle 600`	IPv4 または IPv6、または Dual-stack の加入者は、アイドル タイムアウト機能をサポートします。 （注） 動的なテンプレートまたは RADIUS からの通過の下で設定されたタイマーが作動すると、IPoE セッションに設定されたアイドル タイムアウトは、加入者セッションをクリアすることが予想されます。

	コマンドまたはアクション	目的
ステップ 5	**accounting aaa list default type session periodic-interval** *value* **dual-stack-delay** *value* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# accounting aaa list default type session periodic-interval 60 dual-stack-delay 1`	加入者のアカウンティング機能を設定します。

ステップ 6	{**ipv4** |**ipv6**} **mtu** *mtu-bytes*
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 mtu 678`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 mtu 548`
		IPv4 または IPv6 の最大伝送単位（MTU）を設定します。 範囲は 68 ～ 65535 バイトです。 MTU 値は、加入者セッション中に送信できる最大パケット サイズを定義します。

ステップ 7	{**ipv4** |**ipv6**}**verify unicast source reachable-via rx**
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 verify unicast source reachable-via rx`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 verify unicast source reachable-via rx`
		送信元アドレスの到達可能性チェックを実行するパケットの検証に対して uRPF をイネーブルにします。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが

	コマンドまたはアクション	目的
		継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPv4 または IPv6 加入者セッションの動的なテンプレートの作成：例

```
//Creating Dynamic Template for IPv4 Subscriber Session

configure
dynamic-template
type ipsubscriber ipsub1
timeout idle 600
accounting aaa list default type session periodic-interval 60 dual-stack-delay 1
ipv4 mtu 678
ipv4 verify unicast source reachable-via rx
!
!
end

//Creating Dynamic Template for IPv6 Subscriber Session

configure
dynamic-template
type ipsubscriber ipsub1
timeout idle 600
accounting aaa list default type session periodic-interval 60 dual-stack-delay 1
ipv6 mtu 678
ipv6 verify unicast source reachable-via rx
!
!
end
```

IPoE セッション中に実行されるポリシーマップの作成

IPoE 加入者セッション中に事前定義された動的なテンプレートをアクティブ化するポリシーマップを作成するには、次の作業を実行します。 たとえば、このポリシーマップは、動的なテンプレートをアクティブ化し、セッションの開始イベント中にローカルに定義された許可設定を適用します。

手順の概要

1. **configure**
2. **policy-map type control subscriber** *policy_name*
3. **event session-start match-first**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*
6. *sequence_number* **authorize aaa list default format** *format_name* **password** *password*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber IPoE_policy`	名前が「IPoE_policy」でタイプが「control subscriber」の新しいポリシー マップを作成します。
ステップ 3	**event session-start match-first** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first`	アクションを実行するイベント（セッションの開始）を定義します。
ステップ 4	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	加入者が一致する必要があるクラスを設定します。一致があると、障害が見つかるまですべてのアクションを実行します。

	コマンドまたはアクション	目的
ステップ 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ipsub1`	完全な構造のユーザ名を使用して、加入者の認証をトリガーできます。
ステップ 6	*sequence_number* **authorize aaa list default format** *format_name* **password** *password* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 authorize aaa list default format RM_User password Cisco`	加入者のドメイン名を使用して、加入者の許可をトリガーできます。また、完全に構造化されたユーザ名からのドメイン分析を支援するドメイン形式ルールも提供されます。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

IPoE セッション中に実行されるポリシーマップの作成：例

```
configure
policy-map type control subscriber IPoE_policy
```

```
event session-start match-first
class type control subscriber class-default do-until-failure
1 activate dynamic-template ipsub1
1 authorize aaa list default format RM_User password Cisco
!
!
end
```

アクセス インターフェイスでの IPoE 加入者のイネーブル化

アクセス インターフェイスでの IPoE 加入者の作成をイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** *interface-type interface-path-id*
3. **arp learning disable**
4. {**ipv4** |**ipv6**} **address** {*ipv4_address* |*ipv6_address*} *ipsubnet_mask*
5. **service-policy type control subscriber** *policy-name*
6. **encapsulation dot1q** *value*
7. **ipsubscriber** {**ipv4** |**ipv6**}**l2-connected**
8. **initiator dhcp**
9. **initiator unclassified-source**
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundler-Ether400.12`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 • type 引数でインターフェイス タイプを指定します。インターフェイス タイプの詳細については、疑問符（?）オンライン ヘルプ機能を使用してください。

	コマンドまたはアクション	目的
		• instance 引数で物理インターフェイス インスタンスまたは仮想インスタンスを指定します。 ◦物理インターフェイスインスタンスの表記方法は rack/slot/module/port です。値を区切るスラッシュ（/）は、表記の一部として必要です。 ◦仮想インターフェイスインスタンスの数値範囲は、インターフェイスタイプによって異なります。
ステップ 3	**arp learning disable** 例： `RP/0/RSP0/CPU0:router(config-if)# arp learning disable`	アクセスインターフェイスの ARP ラーニングをディセーブルにします。

ステップ 4	{**ipv4** |**ipv6**} **address** {*ipv4_address* |*ipv6_address*} *ipsubnet_mask*
	例：
	`RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.5.1.1 255.255.0.0`
	または
	例：
	`RP/0/RSP0/CPU0:router(config-subif)# ipv6 address 1144:11`
		インターフェイスの IPv4 アドレスまたは IPv6 アドレスを設定します。

ステップ 5	**service-policy type control subscriber** *policy-name* 例： `RP/0/RSP0/CPU0:router(config-subif)# service-policy type control subscriber PL4`	インターフェイスに加入者制御サービス ポリシーを関連付けます。 （注） PL4 ポリシーマップを作成するには、「加入者ポリシーマップの設定」手順を参照してください。
ステップ 6	**encapsulation dot1q** *value* 例： `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 40`	インターフェイスの 802.1Q フレーム入力を適切なサービス インスタンスにマップするための一致基準を定義します。 値の範囲は 1 ～ 4094 です。
ステップ 7	**ipsubscriber** {**ipv4** |**ipv6**}**l2-connected** 例： `RP/0/RSP0/CPU0:router(config-subif)# ipsubscriber ipv4 l2-connected` または	サブインターフェイスでの L2 接続型 IPv4 または IPv6 加入者の作成をイネーブルにします。

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router(config-subif)# ipsubscriber ipv6 l2-connected`	

ステップ 8	**initiator dhcp**
	
	例：
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv4-l2conn)# initiator dhcp`
	または
	
	例：
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv6-l2conn)# initiator dhcp`
		発信元が分類されていない発信側と組み合わせることもできるアクセスインターフェイスでの DHCP に基づく IPoE 加入者の作成をイネーブルにします。

ステップ 9	**initiator unclassified-source**
	
	例：
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv4-l2conn)# initiator unclassified-source`
	または
	
	例：
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv6-l2conn)# initiator unclassified-source`
		DHCP によって開始された IP セッション用の、DHCP のクラスを開始します。
		
		（注） 複数の発信側が使用されている場合、異なる発信元の IP アドレスの重複を防ぐためにポリシーマップまたはクラス マップを使用します。

ステップ 10	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッショ

	コマンドまたはアクション	目的
		ンは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アクセス インターフェイスでの IPoE 加入者のイネーブル化：例

```
configure
interface Bundler-Ether400.12
arp learning disable
ipv4 address 3.5.1.1 255.255.0.0
service-policy type control subscriber PL4
encapsulation dot1q 40
ipsubscriber ipv4 l2-connected
initiator dhcp
initiator unclassified-source
!
!
end

configure
interface Bundler-Ether400.12
arp learning disable
ipv6 address 4444:34
service-policy type control subscriber PL4
encapsulation dot1q 40
ipsubscriber ipv6 l2-connected
initiator dhcp
initiator unclassified-source
!
!
end
```

PPPoE セッションの確立

PPP プロトコルは、クライアントやサーバなど、2 つのノード間の通信に主に使用されます。 PPP プロトコルは、ポイントツーポイント リンク上でマルチプロトコルの図を転送するための標準的な方式を提供します。 PPP リンクを介して送信できるさまざまなネットワーク プロトコルについて、カプセル化方式、リンク層制御プロトコル（LCP）、および一連のネットワーク制御プロトコル（NCP）を定義します。 LCP は、データ リンクを設定および維持するために使用されます。 PPP ピアは、LCP を使用して、さまざまなリンク層のプロパティまたは特性をネゴシエートできます。 NCP は、プロトコルのデータ パケットを送信する前に、関連付けられているネットワーク プロトコルを確立および設定するために使用されます。

PPP 接続を確立する方法の 1 つは、PPP over Ethernet（PPPoE）の使用です。 PPPoE セッションでは、ポイントツーポイント プロトコル（PPP）は CPE と BNG の間で動作します。 （CPE の一部である）ホーム ゲートウェイは、BNG で終端する PPP ヘッダー（カプセル化）を追加します。

CPE は、次に示すさまざまな PPPoE Active Discovery（PAD）メッセージを使用して BNG を検出し、対話します。

- PPPoE Active Discovery Initiation（PADI）：CPE が、BNG を検出するプロセスを開始するためにブロードキャストします。
- PPPoE Active Discovery Offer（PADO）：BNG はオファーを返します。
- PPPoE Active Discovery Request（PADR）：CPE が、接続を確立するように要求します。
- PPPoE Active Discovery Session confirmation（PADS）：BNG は要求を受け入れ、セッション ID（Session-ID）の割り当てによって応答します。
- PPPoE Active Discovery Termination（PADT）：CPE または BNG が、セッションを終了します。

PPPoE クライアントが複数の BNG に接続された冗長な BNG セットアップでは、CPE によって送信される PADI メッセージはすべての BNG で受信されます。 各 BNG は、PADO メッセージで応答します。 加入者が複数の BNG セットアップで BNG の 1 つを選択できるようにするには、BNG でスマート サーバ選択を設定する必要があります。 PPPoE スマート サーバ選択, （138 ページ）を参照してください。

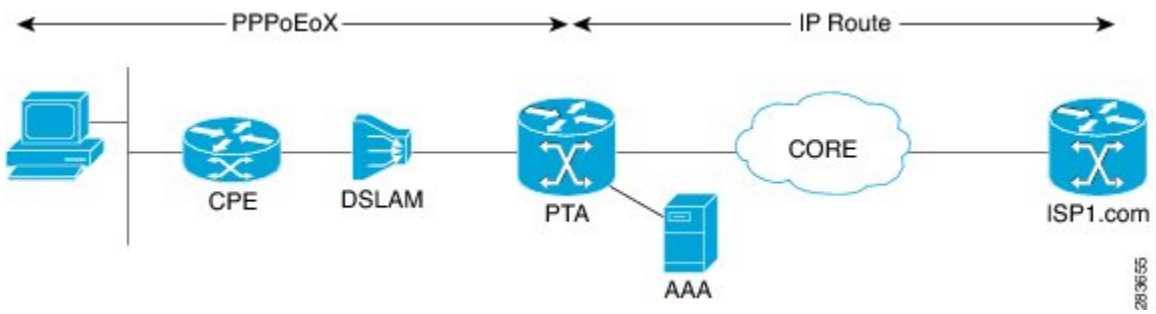
BNG は、さまざまなパラメータに基づいて、PPPoE セッションの要求数を制限および抑えるために、設定の柔軟性を提供します。 詳細については、PPPoE セッション制限, （141 ページ）および PPPoE セッション スロットル, （144 ページ）を参照してください。

PPPoE セッションには、PPP PTA と PPP LAC の 2 つのタイプがあります。 PPP PTA と PPP LAC セッションの機能について、必要に応じてセッションを認証し、転送するように RADIUS サーバを設定する必要があります。 BNG では、使用可能なローカル認証はありません。 PPP PTA と PPP LAC セッションについては、PPP PTA セッションのプロビジョニング, （107 ページ）と PPP LAC セッションのプロビジョニング, （116 ページ）の項で説明します。

PPP PTA セッションのプロビジョニング

PPP Termination and Aggregation（PTA）セッションでは、PPP カプセル化は BNG で終了します。終了後、BNG は IP ルーティングを使用してトラフィックをサービス プロバイダーにルーティングします。 一般的な PTA セッションを次の図に示します。

図 *6*： *PTA* セッション



PPPoE セッションの設定情報は、PPPoE プロファイルに含まれています。 プロファイルを定義した後、アクセス インターフェイスに割り当てることができます。 複数の PPPoE プロファイルを作成し、複数のインターフェイスに割り当てることができます。 グローバルな PPPoE プロファイルも作成できます。グローバルなプロファイルは、特定の PPPoE プロファイルが割り当てられていない任意のインターフェイスのデフォルト プロファイルとして機能します。

PTA PPP セッションは、通常、同じサービス オペレータが加入者にブロードバンド接続を提供し、ネットワーク サービスも管理するネットワーク サービス プロバイダー（小売）モデルで使用されます。 PPP PTA セッションのプロビジョニング プロセスは、次のとおりです。

- PPPoE セッションの PPPoE プロファイルの作成。 PPPoE プロファイルの作成, （108 ページ）を参照してください。
- PPPoE セッションのさまざまな設定を含む動的なテンプレートの作成。 PPP Dynamic-Template の作成, （109 ページ）を参照してください。
- 動的なテンプレートをアクティブ化するポリシーマップの作成。 PPPoE セッション中に実行されるポリシーマップの作成, （111 ページ）を参照してください。
- 加入者作成のイネーブル化。アクセス インターフェイスに PPPoE プロファイルとサービス ポリシーを適用します。 アクセス インターフェイスへの PPPoE 設定の適用, （114 ページ）を参照してください。

  加入者作成機能は、BNG で明示的にイネーブルである必要があります。 この機能がイネーブルでない場合、システムは加入者を分類しようとしません。 結果として、パケットは着信インターフェイス モードに基づいて転送されます。

PPPoE プロファイルの作成

PPPoE プロファイルを作成するには、次の作業を実行します。 PPPoE プロファイルは、後でアクセス インターフェイスに適用されます。

手順の概要

1. **configure**
2. **pppoe bba-group** *bba-group name*
3. **service name** *service_name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pppoe bba-group** *bba-group name* 例： `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	ユーザが指定した名前で PPPoE プロファイルを作成します。
ステップ 3	**service name** *service_name* 例： `RP/0/RSP0/CPU0:router(config-bbagroup)# service name service_1`	加入者が要求したサービスを示します。 加入者プロファイルに追加するサービス名ごとにこの手順を繰り返します。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	レーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

PPPoE プロファイルの作成：例

```
configure
pppoe bba-group bba_1
service name service_1
!
!
end
```

PPP Dynamic-Template の作成

PPP Dynamic-Template を作成するには、次の作業を実行します。 たとえば、この動的なテンプレートは、PAP および CHAP 認証方式を適用するために作成されます。

手順の概要

1. **configure**
2. **dynamic-template type ppp** *dynamic_template_name*
3. **ppp authentication pap**
4. **ppp authentication chap**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template type ppp** *dynamic_template_name* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp ppp_pta_template`	PPPセッションのユーザ定義名を使用して動的なテンプレートを作成します。
ステップ 3	**ppp authentication pap** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication pap`	リンク制御プロトコル（LCP）によるリンクネゴシエーション中に PAP タイプの認証の使用をイネーブルにします。
ステップ 4	**ppp authentication chap** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication chap`	リンク制御プロトコル（LCP）によるリンクネゴシエーション中に CHAP タイプの認証の使用をイネーブルにします。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション

	コマンドまたはアクション	目的
		セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPP Dynamic-Template の作成：例

```
configure
dynamic-template type ppp ppp_pta_template
ppp authentication pap
ppp authentication pap chap
!
!
end
```

PPPoE セッション中に実行されるポリシーマップの作成

PPPoE 加入者セッション中に PPP Dynamic-Template をアクティブ化するポリシーマップを作成するには、次の作業を実行します。 たとえば、このポリシーマップは、セッションの開始イベント中に動的なテンプレートをアクティブ化します。 また、このポリシーマップは、セッションのアクティブ化イベント中にローカルに定義された許可設定も適用します。

手順の概要

1. **configure**
2. **policy-map type control subscriber** *policy_name*
3. **event session-start match-all**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*
6. **event session-activate match-all**
7. **class type control subscriber** *class_name* **do-until-failure**
8. *sequence_number* **authenticate aaa list default**
9. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber PPPoE_policy`	ユーザ定義名が「PPPoE_policy」でタイプが「control subscriber」の新しいポリシー マップを作成します。
ステップ 3	**event session-start match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	アクションを実行するイベント（セッションの開始）を定義します。
ステップ 4	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber pta_class do-until-failure`	加入者が一致するクラスを設定します。一致があると、障害が見つかるまですべてのアクションを実行します。
ステップ 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ppp_pta_template`	指定した動的なテンプレート名で動的なテンプレートをアクティブ化します。
ステップ 6	**event session-activate match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-activate match-all`	アクションを実行するイベント（セッションのアクティブ化）を定義します。
ステップ 7	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber PPP_class do-until-failure`	加入者が一致するクラスを設定します。一致があると、障害が見つかるまですべてのアクションを実行します。

	コマンドまたはアクション	目的
ステップ 8	*sequence_number* **authenticate aaa list default** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 authenticate aaa list default`	完全な構造のユーザ名を使用して、加入者の認証をトリガーできます。

ステップ 9	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPPoE セッション中に実行されるポリシーマップの作成：例

```
configure
policy-map type control subscriber policy1
event session-start match-all
class type control subscriber pta_class do-until-failure
1 activate dynamic-template template1
!
!
event session-activate match-all
class type control subscriber pta_class1 do-until-failure
1 activate dynamic-template ppp_pta_template
end-policy-map
```

アクセス インターフェイスへの PPPoE 設定の適用

アクセスインターフェイスにPPPoEプロファイルとポリシーマップを適用するには、次の作業を実行します。次の作業を完了すると、インターフェイス上でPPPoEトラフィックを受信できるようになります。

はじめる前に

PPPoE プロファイルの作成，（108 ページ）を実行してから、この作業を実行する必要があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type control subscriber** *policy_name*
4. **pppoe enable bba-group** *bbagroup_name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 5.1`	バンドルインターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**service-policy type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	インターフェイスに加入者制御サービス ポリシーを関連付けます。

	コマンドまたはアクション	目的
ステップ 4	**pppoe enable bba-group** *bbagroup_name* 例： `RP/0/RSP0/CPU0:router(config-if)# pppoe enable bba-group bba_1`	Bundle-Ether インターフェイス上で PPPoE をイネーブルにし、このインターフェイス上で使用される bba_1 という名前の PPPoE プロファイルを指定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

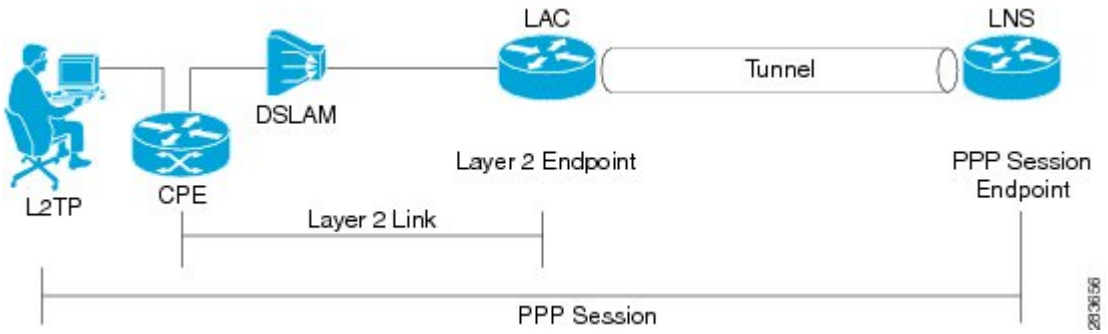
アクセス インターフェイスへの PPPoE 設定の適用：例

```
configure
interface Bundle-Ether100.10
service-policy type control subscriber PL1
pppoe enable bba-group bba_1
!
!
end
```

PPP LAC セッションのプロビジョニング

PPP LAC セッションでは、PPP セッションは、レイヤ 2 トンネル プロトコル（L2TP）を使用し、BNG によってリモート ネットワーク サーバにトンネリングされます。 BNG は、L2TP トンネルに加入者セッションを配置するため、L2TP アクセス コンセントレータ（LAC）のロールを実行します。 トンネルが終端するデバイスは、L2TP ネットワーク サーバ（LNS）と呼ばれます。 PPP LAC セッション中、PPPoE カプセル化は BNG で終了します。ただし、PPP パケットは、L2TP トンネル経由で BNG を越えて LNS に移動します。 一般的な LAC セッションを図 1 に示します。

図 *7*：*LAC* セッション



PPP LAC セッションは、アクセス ネットワーク プロバイダー（卸し売り）モデルで使用されます。ここでは、ネットワーク サービス プロバイダー（NSP）は、ローカル アクセス ネットワーク プロバイダー（ANP）の別個のエンティティです。 NSP は、アクセス認証を実行し、IP アドレスを管理して加入者に提供し、全体的なサービスを担当します。 ANPは、カスタマーへのラストマイル デジタル接続の提供と、加入者トラフィックでの NSP への受け渡しを担当します。 このタイプのセットアップでは、ANP は LAC を所有し、NSP は LNS を所有します。

PPP LAC セッションは、サービス プロバイダー ネットワークの加入者デバイスとノード間で仮想ポイントツーポイント接続を確立します。 加入者は、近くの L2TP アクセス コネクタ（LAC）にダイヤルします。 トラフィックは、サービス プロバイダー ネットワークに存在する LNS にトンネルを介して安全に転送されます。 この全体的な導入アーキテクチャは、バーチャルプライベート ダイヤルアップ ネットワーク（VPDN）としても知られています。

フラグメント化された L2TP データ パケットの再構築は、これらのパケットがドロップされないように、LAC でイネーブル化されます。 LAC での L2TP の再構築, （117 ページ）を参照してください。

PPP LAC セッションは、ノンストップ ルーティング（NSR）とともにステートフル スイッチオーバー（SSO）をサポートし、RP フェールオーバー中のトラフィック損失を削減します。 詳細については、LAC SSO, （119 ページ）を参照してください。

PPP LAC セッションのプロビジョニング プロセスは、次のとおりです。

- 特定の VPDN 設定を使用したテンプレートの定義。 VPDN テンプレートの設定, （124 ページ）を参照してください。

- 同時に確立できる VPDN セッションの最大数の定義。 最大同時 VPDN セッションの設定, （126 ページ）を参照してください。
- VPDN イベント メッセージのロギングのアクティブ化。 VPDN ロギングのアクティブ化, （128 ページ）を参照してください。
- 発信側ステーション ID の適用方法の指定。 発信側ステーション ID に適用するオプションの設定, （130 ページ）を参照してください。
- Session-ID の指定。 L2TP Session-ID コマンドの設定, （132 ページ）を参照してください。
- L2TP クラスに特定の設定の定義。 L2TP クラス オプションの設定, （133 ページ）を参照してください。
- 追加の VPDN セッションの作成の防止。 VPDN の Softshut の設定, （137 ページ）を参照してください。

LAC での L2TP の再構築

L2TP アクセス コンセントレータ（LAC）での L2TP の再構築機能は、LAC と L2TP ネットワーク サーバ（LNS）間の仲介ネットワークでフラグメント化された L2TP データ パケットを再構築します。 データ パケットは、IPv4 コアの最大伝送単位（MTU）を超過するとフラグメント化されます。 この機能をイネーブルにすると、フラグメント化されたパケットがドロップされなくなり、これらのデータ パケットが引き続き転送されます。

L2TP の再構築機能が LAC でディセーブルの場合、フラグメント化されたデータ パケットはドロップされます。 機能は非 L2TP パケットの再構築には影響しません。 ロード バランシングが各パケットに対して発生するかどうかに関係なく、非 L2TP アプリケーションのパケットが適切に再構築されていることを確認するには、次を推奨します。

- 別個のループバック アドレスは、L2TP トラフィック専用に設定されます。 ルータの他のアプリケーションは、この IP アドレスを使用しません。
- 複数のループバック アドレスが L2TP に使用されますが、すべての VRF の他のアプリケーションは、これらのアドレスを使用しません。

再構成のエラーや、フラグメンテーション タイムアウトの場合、ルート スイッチ プロセッサ（RSP）に転送される前にトラフィック フローが保持される最大時間は 250 ミリ秒です。

制約事項

L2TP の再構築機能のイネーブル化は、次の制約事項に従います。

- Typhoon ラインカードのみが、L2TP の再構築機能をサポートします
- IPv4 のフラグメント化されたパケットのみが再構築されます
- 2 つのフラグメントを持つパケットのみが再構築されます
- フラグメントは重複できません
- フラグメント化された IP ヘッダーにオプションを含めることはできません

- フラグメント化された L2TP パケットは、同じラインカードに渡される必要があります。 つまり、仲介ネットワークは、異なるラインカードにフラグメントを到着させるパケットのロードバランシング方式に従って使用することはできません。 一方、非 L2TP パケットの再構築はパケットが異なるラインカードに到着する場合でも影響を受けません。

LAC での L2TP の再構築のイネーブル化

L2TP アクセス コンセントレータ（LAC）での L2TP の再構築をイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn**
3. **l2tp reassembly**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN コンフィギュレーション モードを開始します。
ステップ 3	**l2tp reassembly** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp reassembly`	LAC での L2TP の再構築をイネーブルにします。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

LAC での L2TP の再構築のイネーブル化：例

```
configure
vpdn
l2tp reassembly
!
end
```

LAC SSO

L2TP アクセス コンセントレータ ステートフル スイッチオーバー（LAC SSO）機能は、RP の 1 つをアクティブなプロセッサとして確立し、他の RP をスタンバイ プロセッサとして指定し、それらの間で重要なステート情報を同期します。 デュアル RP をサポートする特定のシスコ ネットワーキング デバイスでは、LAC SSO は RP の冗長性を活用してネットワークのアベイラビリティを向上させます。

LAC SSO は、RP フェールオーバーの場合に、VPDN と L2TP プロトコルのノンストップ ルーティング（NSR）をサポートします。 NSR は、アクティブ RP とスタンバイ RP 間で信頼できる L2TP と VPDN の同期を保証する機能を提供します。 RP フェールオーバーの場合、すべての VPDN および L2TP トンネルの情報とセッション情報が、L2TP ネットワーク ピアに影響を与えることなく保持されます。 また、ピア ネットワーキング デバイスでルーティング フラップが発生することがなくなるため、カスタマーに対するサービス停止を回避できます。 VPDN と LAC SSO がイネーブルな場合、すべてのトンネルとセッションがバックアップ RP にミラーリングされます。

プロセス障害時の RPFO のイネーブル化

アプリケーションまたはプロセスのクラッシュ時に、VPDN NSR がイネーブルの場合、RP フェールオーバーがトリガーされ、新しいプライマリ RP プロセスがトラフィックを損失することなく再起動します。

VPDN NSR は、デフォルトではディセーブルになっています。 RPFO をイネーブルにするには、次の手順を実行します。

手順の概要

1. **configure**
2. **nsr process-failures switchover**
3. **vpdn**
4. **redundancy**
5. **process-failures switchover**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**nsr process-failures switchover** 例： `RP/0/RSP0/CPU0:router(config)# l2tp nsr process-failures switchover`	VPDN ノンストップ ルーティングをイネーブルにします。
ステップ 3	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN コンフィギュレーション モードを開始します。
ステップ 4	**redundancy** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# redundancy`	VPDN 冗長コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 5	**process-failures switchover** 例： `RP/0/RSP0/CPU0:router(config-vpdn-redundancy)# process-failures switchover`	プロセス障害時に、スイッチオーバーを強制します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

LAC SSO のイネーブル化

LAC/VPDN SSO 機能をイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn**
3. **redundancy**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. **show vpdn redundancy**
6. **show vpdn redundancy mirroring**
7. **show l2tpv2 redundancy**
8. **show l2tpv2 redundancy mirroring**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN コンフィギュレーション モードを開始します。
ステップ 3	**redundancy** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# redundancy`	VPDN 冗長コンフィギュレーション モードを開始します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show vpdn redundancy** 例： `RP/0/RSP0/CPU0:router# show vpdn redundancy`	すべての VPDN 冗長の関連情報を表示します。
ステップ 6	**show vpdn redundancy mirroring** 例： `RP/0/RSP0/CPU0:router# show vpdn redundancy mirroring`	VPDN に関連するミラーリング統計情報を表示します。
ステップ 7	**show l2tpv2 redundancy** 例： `RP/0/RSP0/CPU0:router# show l2tpv2 redundancy`	L2TP 冗長の関連情報を表示します。
ステップ 8	**show l2tpv2 redundancy mirroring** 例： `RP/0/RSP0/CPU0:router# show l2tpv2 redundancy mirroring`	L2TP に関連するミラーリング統計情報を表示します。

VPDN SSO のイネーブル化：例

```
configure
 vpdn
  redundancy
   process-failures switchover
end
```

VPDN テンプレートの設定

VPDN テンプレートを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn template**
3. **l2tp-class** *class_name*
4. **tunnel busy timeout** *timeout_value*
5. **caller-id mask-method remove match** *match_substring*
6. **dsl-line-info-forwarding**
7. **ip tos** *type_of_service_value*
8. **vpn id** *value*
9. **vpn vrf** *vrf_name*
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn template** 例： `RP/0/RSP0/CPU0:router(config)# vpdn template`	VPDN テンプレート サブモードを開始します。
ステップ 3	**l2tp-class** *class_name* 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# l2tp-class class_temp`	l2tp class コマンドを設定します。
ステップ 4	**tunnel busy timeout** *timeout_value* 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# tunnel busy timeout 456`	l2tp tunnel busy list コマンドを設定します。 ビジー状態のタイムアウト値は 60 ～ 65535 です。

	コマンドまたはアクション	目的
ステップ 5	**caller-id mask-method remove match** *match_substring* 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# caller-id mask-method remove match m1`	指定された一致サブ文字列で文字をマスキングして発信側ステーション ID に適用するオプションを設定します。
ステップ 6	**dsl-line-info-forwarding** 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# dsl-line-info-forwarding`	DSL回線情報属性を転送します。
ステップ 7	**ip tos** *type_of_service_value* 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# ip tos 56`	トンネリングされたトラフィックの IP ToS 値を設定します。 サービス値の範囲は 0～255 です。
ステップ 8	**vpn id** *value* 例： `RP/0/RSP0/CPU0:router(config-vpdn-temp)# vpn id 3333:33`	VPN のトンネルを指定し、値 3333:33 で VPN ID を設定します。 値の範囲は 16 進数で 0 ～ ffffff です。
ステップ 9	**vpn vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:router(config-vpdn-template)# vpn vrf vrf_1`	VPN VRF 名を設定します。

ステップ 10	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

VPDN テンプレートの設定：例

```
configure
l2tp-class class hello-interval 100
vpdn
template l2tp-class class //template default will be used and display in show run
template tunnel busy timeout 567
l2tp-class class

vpdn
template default
l2tp-class class
!
end
```

最大同時 VPDN セッションの設定

トンネルごとに限定してセッションの最大同時 VPDN セッションを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn**
3. **session-limit** *number_of_sessions*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN をイネーブルにして、VPDN サブモードを開始します。
ステップ 3	**session-limit** *number_of_sessions* 例： `RP/0/RSP0/CPU0:router(config-vpdn)# session-limit 200`	最大同時 VPDN セッションを設定します。 範囲は 1 ～ 131072 です。 （注） 一部のセッションの起動後に制限が設定されている場合、これらのセッションは制限に関係なく動作し続けます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC

	コマンドまたはアクション	目的
		モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

最大同時 VPDN セッションの設定：例

```
configure
vpdn
session-limit 200
!
end
```

VPDN ロギングのアクティブ化

VPDN イベント情報のロギングをアクティブ化するには、次の作業を実行します。 VPDN イベントのロギングをイネーブルにすると、VPDN イベント メッセージは、イベントの発生時にロギングされます。

（注） トンネルの開始レコードと終了レコードは、トンネル統計情報なしで生成されます。

手順の概要

1. **configure**
2. **vpdn**
3. **logging** [**cause**| **cause-normal** | **dead-cache** | **local** | **tunnel-drop** | **user**]
4. **history failure**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN サブモードを開始します。

ステップ 3	**logging** [**cause**| **cause-normal**	**dead-cache**	**local**	**tunnel-drop**	**user**]
					
					例：
					
					`RP/0/RSP0/CPU0:router(config-vpdn)# logging local`
					`RP/0/RSP0/CPU0:router(config-vpdn)# logging user`
					`RP/0/RSP0/CPU0:router(config-vpdn)# logging cause`
					`RP/0/RSP0/CPU0:router(config-vpdn)# logging tunnel-drop`
						一般的な VPDN イベントのロギングをイネーブルにします。

ステップ 4	**history failure** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# history failure`	履歴障害テーブルへの VPDN 障害イベントのロギングをイネーブルにします。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュ

	コマンドまたはアクション	目的
		レーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

VPDN ロギングのアクティブ化：例

```
configure
vpdn
history failure
logging local
logging user
logging cause-normal
logging tunnel-drop
logging dead-cache
!
end
```

発信側ステーション ID に適用するオプションの設定

発信側ステーション ID に適用するオプションを設定するには、次の作業を実行します。 発信側ステーション ID は、発信者の電話番号、LAC での接続に使用される論理回線 ID（LLID）、またはネットワークへの PC 接続の MAC アドレスなど、セッション発信者に関する詳細情報を提供します。

手順の概要

1. **configure**
2. **vpdn**
3. **caller-id mask-method remove match** *match_name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN サブモードを開始します。
ステップ 3	**caller-id mask-method remove match** *match_name* 例： `RP/0/RSP0/CPU0:router(config-vpdn)# caller-id mask-method remove match match_class`	すべてのユーザの発信側ステーション ID を抑制します。「match」オプションがある場合、ユーザ名に「match-string」があるユーザの発信側ステーション ID のみが抑制されます。 （注） また、このコマンドは、VPDN テンプレート コンフィギュレーション モードでも実行されます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

発信側ステーション ID に適用するオプションの設定：例

```
configure
vpdn //or vpdn template
caller-id mask-method remove match match_call
!
end
```

L2TP Session-ID コマンドの設定

L2TP session-id コマンドを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn**
3. **l2tp session-id space hierarchical**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN を設定します。
ステップ 3	**l2tp session-id space hierarchical** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp session-id space hierarchical`	階層型 Session-ID 割り当てアルゴリズムをイネーブルにします。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

L2TP Session-ID コマンドの設定：例

```
configure
vpdn
l2tp session-id space hierarchical
!
end
```

L2TP クラス オプションの設定

L2TP クラスのさまざまなオプションを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **l2tp-class** *class_name*
3. **authentication** [**disable** | **enable**]
4. **congestion control**
5. **digest** [**check disable** |**hash** { **MD5** | **SHA1** } | **secret** { **0** | **7** | **LINE** }]
6. **hello-interval** *interval_duration*
7. **hostname** *host_name*
8. **receive-window** *size*
9. **retransmit initial** [**retries** | *retries_number* | **timeout** {**max** *max_seconds* | **min** *min_seconds* }
10. **timeout** [**no-user**{ *timeout_value* | **never** } | **setup** *setup_value*]
11. **tunnel accounting** *accounting_method_list_name*
12. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2tp-class** *class_name* 例： `RP/0/RSP0/CPU0:router(config)# l2tp-class class1`	L2TP class コマンドを設定します。
ステップ 3	**authentication** [**disable**	**enable**] 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# authentication disable`	トンネル認証をイネーブルにします。 Enable および Disable オプションは、L2TP トンネル認証をイネーブルまたはディセーブルにします。
ステップ 4	**congestion control** 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# congestion control`	L2TP の輻輳制御をイネーブルにします。

	コマンドまたはアクション	目的

ステップ 5	**digest** [**check disable** |**hash** { **MD5**	**SHA1** }	**secret** { **0**	**7**	**LINE** }]
					
					例：
					`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest check disable`
					`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest hash MD5`
					`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest secret 0`
						L2TPv3 制御接続のダイジェスト設定をメッセージとして送ります。

ステップ 6	**hello-interval** *interval_duration* 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# hello-interval 45`	秒単位で指定された HELLO メッセージ間隔を設定します。
ステップ 7	**hostname** *host_name* 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# hostname local_host`	制御接続の認証に対するローカル ホスト名を設定します。
ステップ 8	**receive-window** *size* 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# receive-window 56`	制御接続のウィンドウ サイズを受け取ります。 範囲は 1 ～ 16384 です。

ステップ 9	**retransmit initial** [**retries**	*retries_number*	**timeout** {**max** *max_seconds*	**min** *min_seconds* }
				
				例：
				`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit initial retries 58`
				`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit initial timeout max 6`
					制御接続のウィンドウ サイズを受け取ります。 範囲は 1 ～ 16384 です。

ステップ 10	**timeout** [**no-user**{ *timeout_value*	**never** }	**setup** *setup_value*]
			
			例：
			`RP/0/RSP0/CPU0:router(config-l2tp-class)# timeout no-user 56`
			`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit setup 60`
				制御接続のウィンドウ サイズを受け取ります。 タイムアウト値の範囲は、秒単位で 0 ～ 86400 です。 設定値の範囲は 60 ～ 6000 です。

ステップ 11	**tunnel accounting** *accounting_method_list_name* 例： `RP/0/RSP0/CPU0:router(config-l2tp-class)# tunnel`	AAAアカウンティング方式リストの名前を設定します。

	コマンドまたはアクション	目的
	`accounting acc_tunn`	

ステップ 12	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

L2TP クラス オプションの設定：例

```
configure
l2tp-class class1
authentication enable
congestion-control
digest check disable
hello-interval 876
hostname l2tp_host
receive-window 163
retransmit initial timeout 60
timeout no-user 864
```

```
tunnel accounting aaa_l2tp
!
end
```

VPDN の Softshut の設定

VPDN の softshut を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **vpdn**
3. **softshut**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN サブモードを開始します。
ステップ 3	**softshut** 例： `RP/0/RSP0/CPU0:router(config-vpdn)# softshut`	新しいセッションが許可されていないことを確認します。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	レーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

VPDN の Softshut の設定：例

```
configure
vpdn
softshut
!
end
```

PPPoE スマート サーバ選択

BNG の PPPoE スマート サーバ選択（PADO 遅延）機能によって、PPPoE クライアントは、マルチ BNG セットアップで、セッションを確立するために BNG の選択を制御できます。 この機能は、PPPoE クライアントから受信した PADI メッセージに応じて、BNG から PADO メッセージを送信する場合の遅延を設定するオプションを提供します。 これは、すべての BNG で優先順位とロード バランシングの確立に役立ちます。

マルチ BNG のセットアップで PPPoE セッションを確立すると、クライアントはすべての BNG に PADI メッセージをブロードキャストします。 BNG が PADO メッセージで応答すると、加入者は BNG を選択し、セッションを確立する必要がある BNG に PADR にメッセージを送信します。 ほとんどの PPPoE クライアントは、最初に PADO メッセージを受信した BNG に PADR にメッセージを送信します。 スマート サーバ選択機能を BNG で設定すると、PPPoE クライアントから受信した PADI メッセージのプロパティに基づいて、BNG から送信された PADO メッセージに遅延が追加されます。 PADO パケットの受信時のこの遅延は、PPPoE クライアントに、PADR メッセージが送信される適切な BNG を効果的に選択できる柔軟性を与えます。

スマート サーバ選択のオプションの設定

- BNG から送信された PADO メッセージの固有の遅延を設定できます。

- 着信 PADI メッセージに含まれる Circuit-ID、Remote-ID、および Service-Name に基づいて、BNG から送信される PADO メッセージの遅延を設定できます。
- Circuit-ID タグおよび Remote-ID タグと、長さ 64 文字までの文字列との一致を許可します。
- 着信 PADI メッセージに含まれる Circuit-ID、Remote-ID、および Service-Name の部分一致を許可します。

PADO メッセージの遅延の設定については、PADO 遅延の設定, （139 ページ）を参照してください。

PADO 遅延の設定

PPPoE Active Discovery Offer（PADO）メッセージの遅延を設定する、つまり、BNG の PPPoE BBA グループのスマート サーバ選択機能をイネーブルにするには、次の作業を実行します。

（注） 複数の遅延が特定の加入者と一致する場合、Circuit-ID の一致は Remote-ID の一致よりも優先され、同様に Remote-ID の一致は Service-Name の一致よりも優先されます。

手順の概要

1. **configure**
2. **pppoe bba-group** *bba-group-name*
3. 特定の遅延値、Circuit-ID、Remote-ID、および Service-Name のそれぞれに基づいて PADO 遅延を設定するには、次のコマンドを使用します。
   - **pado delay** *delay*
   - **pado delay circuit-id** {*delay* | {**string** | **contains**} *string delay*}
   - **pado delay remote-id** {*delay* | {**string** | **contains**} *string delay*}
   - **pado delay service-name** {**string** | **contains**} *string delay*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pppoe bba-group** *bba-group-name* 例： `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	PPPoE BBA-Group コンフィギュレーション モードを開始します。

ステップ 3	特定の遅延値、Circuit-ID、Remote-ID、およびService-Nameのそれぞれに基づいてPADO遅延を設定するには、次のコマンドを使用します。
	• **pado delay** *delay*
	• **pado delay circuit-id** {*delay* | {**string** | **contains**} *string delay*}
	• **pado delay remote-id** {*delay* | {**string** | **contains**} *string delay*}
	• **pado delay service-name** {**string** | **contains**} *string delay*
	例：
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay 500`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay circuit-id 200`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay remote-id string circuit4`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay service-name contains service 9950`
		次の項目に基づいて、ミリ秒単位で PADO 遅延を設定します。
		• 特定の遅延値
		• PADI で受信される Circuit-ID
		• PADI で受信される Remote-ID
		• PADI で受信される Service-Name
		遅延の範囲は 0 ～ 10000 です。
		PADI メッセージで受信した Circuit-ID（または Remote-ID またはService-Name）が設定した *string* 値と一致すると、**string** オプションによって PADO メッセージが遅延します。
		PADI メッセージで受信した Circuit-ID（または Remote-ID またはService-Name）に設定した *string* 値が含まれると、**contains** オプションによって PADO メッセージが遅延します。

ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーショ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	ン セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPPoE PADO 遅延の設定：例

```
 pppoe bba-group bba_1
 pado delay 500
 pado delay remote-id 100
 pado delay circuit-id string circuit4 8000
 pado delay service-name contains service 9950
!
end
```

PPPoE セッション制限およびスロットル

PPPoE セッション制限

PPPoEセッション制限のサポートは、BNGルータで作成できるPPPoEセッションの数を制限します。 結果として、仮想アクセス用の BNG ルータによる過剰なメモリ使用が削減されます。

これは、次のPPPoEセッション数を制限することによって、BNGルータでの柔軟な追加設定を可能にします。

- ラインカード
- 親インターフェイス
- ピア MAC アドレス
- 個々のアクセス インターフェイス下のピア MAC アドレス
- Circuit-ID

- Remote-ID
- Circuit-ID と Remote-ID の組み合わせ
- 同じ内部 VLAN タグを使用するアクセス インターフェイス
- 同じ外部 VLAN タグを使用するアクセス インターフェイス。
- 同じ内部および外部 VLAN タグを使用するアクセス インターフェイス

PPPoE セッション制限のサポートは、各ピア MAC アドレスと個々のアクセス インターフェイス下の各ピア MAC アドレスに対するインターワーキング機能（IWF）セッション数も制限します。

PPPoE セッション制限の設定, （142 ページ）を参照してください。

PPPoE セッション制限の設定

BNG の PPPoE BBA グループに対する PPPoE セッション制限を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pppoe bba-group** *bba-group name*
3. **sessions** {**access-interface** | **circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {{**mac** | **mac-iwf**} [**access-interface**] }} | **max** | **outer-vlan** | **remote-id** | **vlan**} **limit** *limit-count* [**threshold** *threshold-count*]
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pppoe bba-group** *bba-group name* 例： `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	PPPoE BBA-Group コンフィギュレーション モードを開始します。
ステップ 3	**sessions** {**access-interface**	**circuit-id**	**circuit-id-and-remote-id**	**inner-vlan**	{{**mac**	PPPoE セッション制限を設定します。

	コマンドまたはアクション	目的

	| **mac-iwf**} [**access-interface**] }} | **max** | **outer-vlan** | **remote-id** | **vlan**} **limit** *limit-count* [**threshold** *threshold-count*]
	
	例：
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions access-interface limit 1000`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions mac access-interface limit 5000 threshold 4900`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions circuit-id limit 8000 threshold 7500`
		オプションの引数 **threshold** が設定されている場合、ログ メッセージは、PPPoE セッション制限値が *threshold-count* 値を超過すると生成されます。
		
		limit-count 値と *threshold-count* 値の範囲は 1 ～ 65535 です。デフォルト値は 65535 です。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPPoE セッション制限の設定：例

```
configure
pppoe bba-group bba1
 sessions circuit-id limit 8000 threshold 7500
 sessions access-interface limit 1000
 sessions mac access-interface limit 5000 threshold 900
```

```
!
end
```

PPPoE セッション スロットル

BNG の PPPoE セッション スロットルのサポートは、指定期間内に BNG に着信する PPPoE セッション要求数を制限します。 これにより、BNG サーバに着信する他のクライアント要求のセッションの確立は、影響を受けません。

これは、次のいずれかに基づいて、セッション要求数を抑制することによって、BNGルータでの柔軟な設定を可能にします。

- ピア MAC アドレス
- 個々のアクセス インターフェイス下のピア MAC アドレス
- Circuit-ID
- Remote-ID
- Circuit-ID と Remote-ID の組み合わせ
- 個々のアクセス インターフェイス下の内部 VLAN タグ
- 個々のアクセス インターフェイス下の外部 VLAN タグ
- 個々のアクセス インターフェイス下の内部および外部 VLAN タグ

PPPoE セッション スロットルのサポートは、個々のアクセス インターフェイス下の各ピア MAC アドレスに対するインターワーキング機能（IWF）のセッション要求も抑制します。

PPPoE セッション スロットルの設定, （144 ページ）を参照してください。

PPPoE セッション スロットルの設定

BNG の PPPoE BBA グループに対する PPPoE セッション スロットルを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pppoe bba-group** *bba-group name*
3. **sessions** {**circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {**mac** [**access-interface**] } | {**mac-iwf** {**access-interface**}} | **outer-vlan** | **remote-id** | **vlan**} **throttle** *request-count request-period blocking-period*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pppoe bba-group** *bba-group name* 例： `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	PPPoE BBA-Group コンフィギュレーション モードを開始します。

ステップ 3	**sessions** {**circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {**mac** [**access-interface**] } | {**mac-iwf** {**access-interface**}} | **outer-vlan** | **remote-id** | **vlan**} **throttle** *request-count request-period blocking-period*
	
	例：
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions circuit-id throttle 1000 50 25`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions mac-iwf access-interface throttle 5000 100 50`
		PPPoE セッション スロットルを設定します。
		request-count 値の範囲は 1 ～ 65535 です。
		request-period 値の範囲は 1 ～ 100 です。
		blocking-period 値の範囲は 1 ～ 100 です。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPPoE セッション スロットルの設定：例

```
configure
pppoe bba-group bba1
 sessions circuit-id throttle 1000 50 25
 sessions mac-iwf access-interface throttle 5000 100 50
!
```

DHCP の設定

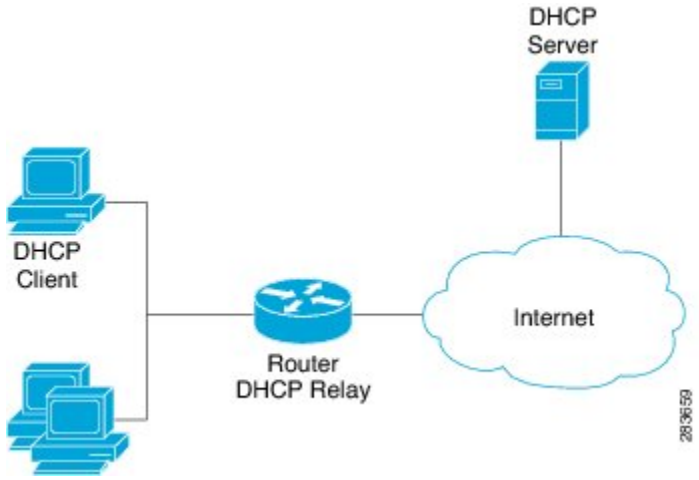
ダイナミック ホスト コンフィギュレーション プロトコル（DHCP）は、IP ネットワーク上で通信できるようにネットワーク デバイスを設定するために使用されるネットワーク プロトコルです。DHCP ネットワークには、3 つの要素があります。

- DHCP クライアント：IP アドレスなどの IP 設定情報を探すデバイスです。
- DHCP サーバ：アドレス プールから DHCP クライアントに IP アドレスを割り当てます。
- DHCP リレーまたは DHCP プロキシ：クライアントとサーバの間で IP 設定情報の受け渡しを行います。 この機能は、DHCP クライアントおよび DHCP サーバが異なるネットワークにある場合に使用されます。

最初、DHCP クライアント（CPE）は IP アドレスを所有していません。 そのため、IP アドレスを取得するために L2 ブロードキャスト要求を送信します。 BNG は、リレー エージェントとして機能し、要求を処理して DHCP サーバに転送します。 BNG は、DHCP サーバから DHCP クライア

ントに返される応答も転送し、エンドデバイスが正しいIP設定情報を取得できるようにします。一般的な DHCP レイアウトを次の図に示します。

図 *8*：*DHCP* ネットワーク



DHCP サーバは、リース期間と呼ばれる設定可能な期間にのみ IP アドレスを割り当てます。 クライアントデバイスがリース期間よりも長い期間IPアドレスを維持する必要がある場合、クライアントは期限前にリースを更新する必要があります。 リースを更新するには、クライアントはDHCP サーバにユニキャスト要求を送信します。 要求メッセージを受信すると、サーバは確認応答で応答し、クライアントのリースが確認応答メッセージで指定されたリース期間まで拡張されます。

コントロール ポリシーがアクセス インターフェイスに適用されると、加入者のアクセス インターフェイスになります。 それ以外の場合は、DHCP のスタンドアロン インターフェイスになります。 スタンドアロン インターフェイスでは、DHCP は設定に基づいて RIB にルートを追加し、ARP エントリを入力します。

加入者のアクセス インターフェイスでは、DHCP はポリシー プレーンを使用して、IP 加入者セッションをクライアント バインディングに対して作成するかどうかを決定します。 これは、有効なコントロール ポリシーをクライアント バインディングが作成されるアクセス インターフェイスに適用するのかどうかに基づいて決定されます。 加入者セッションが作成されると、ルートが加入者インターフェイス用に追加されますが、ARP 要求はその加入者インターフェイスから送信されません。

BNG は、DHCP ネットワークで DHCP リレーまたは DHCP プロキシとして機能するように設定できます。

DHCP リレーのイネーブル化

DHCP リレーとして、BNG は DHCP クライアントのブロードキャストを傍受し、DHCP メッセージに必要な変更を行い、それを DHCP サーバに転送します。 BNG は、クライアントからの DHCP パケットを DHCP サーバに転送するときに、リレー エージェント情報を挿入します。 この情報には、着信回線を識別するための「Circuit-ID」とクライアント MAC アドレスを識別するための

「Remote-ID」が含まれます。 DHCP サーバは、IP アドレッシングとその他のパラメータ割り当てポリシーの実装にリレー エージェント情報を使用します。

DHCP サーバがクライアント要求に応答すると、BNG はクライアントに応答をリレーします。サーバからの DHCP パケットを DHCP クライアントに転送する場合、BNG はサーバが追加したリレー エージェント情報を削除します。 ただし、DHCP サーバの IP アドレスはクライアントに渡されます。 クライアントは、DHCP サーバのアドレスを使用して、サーバにユニキャスト リースの更新要求を直接送信します。

BNG の DHCP リレー エージェントの設定には、次の段階があります。

- リレー プロファイルを作成します。 プロファイルには、さまざまなリレー設定が含まれます。 これらの設定は、プロファイルがインターフェイスに付加されると適用されます。 リレー プロファイルを作成するには、DHCP リレー プロファイルの設定, （148 ページ）を参照してください。
- リレー エージェント情報を設定します。 これらの設定は、リレー プロファイル内で指定されます。 リレー エージェント情報の設定を指定するには、リレー エージェント情報の設定, （150 ページ）を参照してください。

DHCP リレー プロファイルの設定

DHCP リレー エージェントの各種設定を含む新しいリレー プロファイルを作成するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **relay**
4. **helper-address** [**vrf** *vrf- name*] *address*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 コンフィギュレーション サブモードを開始します。
ステップ 3	**profile** *profile-name* **relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	ユーザ定義名で新しいリレープロファイルを作成します。
ステップ 4	**helper-address** [**vrf** *vrf- name*] *address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf vrf1 10.10.1.1`	ダイナミック ホスト コンフィギュレーション プロトコル（DHCP）の IPv4 リレー エージェントを、特定の DHCP サーバに BOOTREQUEST パケットをリレーするように設定します。 • *address* 引数の値には、特定の DHCP サーバアドレスまたはネットワーク アドレス（宛先ネットワーク セグメントに他にも DHCP サーバがある場合）を指定できます。 ネットワーク アドレスを使用することで、他のサーバも DHCP 要求に応答できるようになります。 • サーバが複数ある場合は、各サーバにヘルパー アドレスを 1 つ設定してください。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッ

	コマンドまたはアクション	目的
		ションが終了して、ルータがEXECモードに戻ります。 ◦no と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。 ◦cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

DHCP リレー プロファイルの設定：例

```
configure
dhcp ipv4
profile client relay
helper-address vrf vrf1 10.10.1.1
 !
!
end
```

リレー エージェント情報の設定

リレー エージェント情報を設定するには、次の作業を実行します。 さまざまな設定は、リレー プロファイル内で指定されます。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **relay**
4. **relay information option**
5. **relay information check**
6. **relay information policy** {**drop** | **keep**}
7. **relay information option allow-untrusted**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 コンフィギュレーション サブモードを開始します。
ステップ 3	**profile** *profile-name* **relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 relay`	DHCP IPv4 プロファイル リレー サブモードを開始します。
ステップ 4	**relay information option** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option`	DHCP サーバに転送された BOOTREQUEST メッセージに、BNG が DHCP リレー エージェント情報オプション（オプション 82 フィールド）を挿入できるようにします。 リレー エージェント情報は、サブオプションが 1 つ以上含まれている単一の DHCP オプションとして編成されます。 これらのオプションには、リレー エージェントが認識する情報が含まれています。サポートされるサブオプションは、Remote-ID および Circuit-ID です。

	コマンドまたはアクション	目的
		（注） この機能は、デフォルトではディセーブルになっています。
ステップ 5	**relay information check** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information check`	（任意）転送された BOOTREPLY メッセージ内のリレー エージェント情報オプションの有効性をチェックするように DHCP を設定します。 リレー エージェントは、無効なメッセージを受信すると、そのメッセージをドロップします。有効なメッセージを受信すると、リレー エージェントはリレー エージェント情報オプション フィールドを削除し、パケットを転送します。 デフォルトでは、DHCP は DHCP サーバから受信した DHCP 応答パケットのリレー エージェント情報オプション フィールドの有効性をチェックしません。 （注） 機能がディセーブルになっていた場合、この機能を再度イネーブルにするには **relay information check** コマンドを使用します。
ステップ 6	**relay information policy {drop | keep}** 例： `RP/0/RSP0/CPU0:router(config)# dhcp relay information policy drop`	（任意）DHCP リレー エージェントの再転送ポリシー、つまりリレーエージェントがリレー情報をドロップするのか、保持するのかを設定します。 DHCP リレー エージェントは、デフォルトではリレー情報オプションを置換します。
ステップ 7	**relay information option allow-untrusted** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option allow-untrusted`	（任意）既存のリレー情報オプションがあり、GIADDR（リレーによって切り替えられたゲートウェイ IP アドレス）がゼロに設定されている BOOTREQUEST パケットを廃棄しないように DHCP IPv4 Relay を設定します。
ステップ 8	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存さ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	れ、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

リレー エージェント情報の設定：例

```
configure
dhcp ipv4
profile profile1 relay
relay information option
 !
!
end
```

DHCP プロキシのイネーブル化

DHCP プロキシとして、BNG はリレーのすべての機能を実行し、追加機能も提供します。 プロキシ モードでは、BNG は DHCP クライアントに DHCP サーバの詳細を隠します。 BNG は、クライアントがプロキシをサーバと見なすような DHCP 応答を変更します。 この状態で、クライアントは BNG が DHCP サーバであるかのように対話します。

BNG は、DHCP サーバから IP リースを取得し、プールに保持します。 クライアントがリースを更新する必要がある場合、BNG をサーバと仮定して、リースの更新要求を BNG に直接ユニキャストします。 BNG は、リース プールからリースを割り当てることでリースを更新します。

このように、DHCP プロキシは、次の 2 フェーズにリース管理プロセスを分割します。

- サーバからプロキシ（プロキシ リース）
- プロキシからクライアント（クライアントリース）

2 フェーズのリース管理には、次の機能があります。

- より短いクライアント リース期間とより長いプロキシ リース期間。
- ネットワーク エッジでの高頻度のリース管理（更新）。
- 中央サーバでの低頻度のリース管理（更新）。

DHCP プロキシの長所は、次のとおりです。

- BNG と DHCP サーバ間におけるトラフィックの削減。
- ネットワークの停止に対するより速いクライアント応答。

BNG での DHCP プロキシ設定には、次のフェーズが含まれます。

- プロキシプロファイルの作成。プロファイルには、さまざまなプロキシ設定が含まれます。これらの設定は、プロファイルがインターフェイスに付加されると適用されます。 プロキシ プロファイルを作成するには、DHCP IPv4 プロファイル プロキシ クラスの設定, （154 ページ）を参照してください。
  - クライアント リース期間の指定。 クライアントは、この期間が終了する前にリースを更新する必要があります。そうしなければ、リースが期限切れになります。 プロキシ プロファイル内でクライアント リース期間を指定するには、クライアント リース期間の設定, （160 ページ）を参照してください。
  - Remote-ID の指定。 Remote-ID は、DHCP 要求を送信したホストを識別するためにプロキシによって使用されます。 プロキシ プロファイル内で Remote-ID を定義するには、Remote-ID の設定, （158 ページ）を参照してください。
- インターフェイスの Circuit-ID の指定。 Circuit-ID は、DHCP 要求を受信した回線を識別するためにプロキシによって使用されます。 後で、DHCP プロキシは、適切な回線に DHCP 応答をリレーするために Circuit-ID を使用します。 Circuit-ID は、インターフェイスに定義されます。 定義するには、インターフェイスの Circuit-ID の設定, （156 ページ）を参照してください。
- インターフェイスへのプロキシ プロファイルの接続。 インターフェイスへのプロキシ プロファイルの接続, （161 ページ）を参照してください。

DHCP IPv4 プロファイル プロキシ クラスの設定

DHCP を定義するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **class** *class-name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
6. **show dhcp ipv4 proxy profile name** *name*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	プロキシ プロファイル コンフィギュレーション モードを開始します。 DHCP プロキシは、クラス情報を使用して、特定のプロファイルのパラメータのサブセットを選択します。
ステップ 4	**class** *class-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-profile)# class blue`	DHCP プロキシ プロファイル クラスを作成し、プロキシ プロファイル クラス モードを開始します。
ステップ 5	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	レーションセッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show dhcp ipv4 proxy profile name** *name* 例： `RP/0/RSP0/CPU0:routershow dhcp ipv4 proxy profile name profile1`	（任意）詳細なプロキシ プロファイル情報を表示します。

インターフェイスの Circuit-ID の設定

インターフェイスの Circuit-ID を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **interface** *type interface-path-id*
4. **proxy information option format-type circuit-id** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 コンフィギュレーション サブモードを開始します。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# interface Bundle-Ether 355`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**proxy information option format-type circuit-id** *value* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# proxy information option format-type circuit-id 7`	このインターフェイスの Circuit-ID を設定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

インターフェイスの Circuit-ID の設定：例

```
configure
dhcp ipv4
interface Bundle-Ether100.10
proxy information option format-type circuit-id 7
 !
!
end
```

Remote-ID の設定

Remote-ID を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **relay information option remote-id** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	DHCP プロキシ プロファイルを作成します。
ステップ 4	**relay information option remote-id** *value* 例： `RP/0/RSP0/CPU0:router(config-if)# relay information option remote-id 9`	Remote-ID 値などのリモート ID サブオプションのリレー エージェント情報を挿入します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

Remote-ID の設定：例

```
configure
dhcp ipv4
profile profile1 proxy
relay information option remote-id 9
 !
!
end
```

クライアント リース期間の設定

クライアント リース期間を設定するには、次の作業を実行します。 クライアント リースの期限が切れるまでの期間を定義します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **lease proxy client-lease-time** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	DHCP プロファイルを作成します。
ステップ 4	**lease proxy client-lease-time** *value* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# lease proxy client-lease-time 600`	各プロファイルのクライアント リース期間を設定します。 リース プロキシ クライアント期間の最小値は 300 秒です。
ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

クライアント リース期間の設定：例

```
configure
dhcp ipv4
profile profile1 proxy
lease proxy client-lease-time 600
 !
!
end
```

インターフェイスへのプロキシ プロファイルの接続

インターフェイスにプロキシプロファイルを接続するには、次の作業を実行します。接続後、プロキシ プロファイルに指定されているさまざまな設定がインターフェイスで有効になります。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **interface** *type interface-path-id* **proxy profile** *profile-name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. **show dhcp ipv4 proxy profile name** *name*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type interface-path-id* **proxy profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# interface Bundle-Ether 344 proxy profile profile1`	インターフェイス コンフィギュレーション モードを開始し、インターフェイスにプロキシ プロファイルを割り当てます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show dhcp ipv4 proxy profile name** *name* 例： `RP/0/RSP0/CPU0:router# show dhcp ipv4 proxy profile name profile1`	（任意）詳細なプロキシ プロファイル情報を表示します。

インターフェイスへのプロキシ プロファイルの接続：例

```
configure
dhcp ipv4
interface Bundle-Ether100.10 proxy profile profile1
proxy information option format-type circuit-id 7
 !
!
end
```

DHCP リース制限の指定

DHCP リース制限機能によって、インターフェイスの DHCP バインディング数を制限できます。バインディングは、クライアントの MAC アドレスとクライアントに割り当てられる IP アドレスの間のマッピングを表します。 リース制限は、各 Circuit-ID、Remote-ID、またはインターフェイスに指定できます。

リース制限は、DHCP プロキシ プロファイルを介して設定できます。 このプロファイルをインターフェイスに接続すると、そのインターフェイス上で設定された制限まで、バインディングが許可されます。 たとえば、1 回線あたりのリース制限が 10 バインディングのプロファイルが 4 つのインターフェイスに割り当てられている場合、それぞれの一意の Circuit-ID に対して、インターフェイスごとに 10 バインディングが許可されます。

リース制限が既存のバインディングの現在の数よりも小さい場合、既存のバインディングを持続できますが、バインディングの数が新しいリース制限以下に減少するまで、新しいバインディングを作成することはできません。

リース制限が許可変更（CoA）または Access-Accept メッセージの一部として AAA サーバから指定されている場合、プロキシプロファイルで設定された DHCP リース制限は上書きされます。この場合、AAA サーバから受信した最新のセッション制限は、特定の Circuit-ID の現在のリース制限として使用されます。AAA サーバからのリース制限セットは、リース制限が適用される Circuit-ID に関連付けられたクライアント バインディングがこれ以上ない場合に削除されます。

リース制限を指定するには、次の手順を参照してください。

- Circuit-ID のリース制限の指定，（164 ページ）
- Remote-ID のリース制限の指定，（166 ページ）
- インターフェイスのリース制限の指定，（167 ページ）

Circuit-ID のリース制限の指定

各 Circuit-ID にリース制限を指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-circuit-id** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	DHCP プロファイルを作成します。

	コマンドまたはアクション	目的
ステップ 4	**limit lease per-circuit-id** *value* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-circuit-id 1000`	インターフェイスに適用される Circuit-ID にリース制限を指定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

Circuit-ID のリース制限の指定：例

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-circuit-id 1000
 !
!
end
```

Remote-ID のリース制限の指定

各 Remote-ID にリース制限を指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-remote-id** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	DHCP プロファイルを作成します。
ステップ 4	**limit lease per-remote-id** *value* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-remote-id 1340`	インターフェイスに適用される Remote-ID にリース制限を指定します。
ステップ 5	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		◦ noと入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

Remote-ID のリース制限の指定：例

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-remote-id 1340
 !
!
end
```

インターフェイスのリース制限の指定

各インターフェイスにリース制限を指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-interface** *value*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	IPv4 DHCP コンフィギュレーション モードを開始します。
ステップ 3	**profile** *profile-name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	DHCP プロファイルを作成します。
ステップ 4	**limit lease per-interface** *value* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-interface 2400`	各インターフェイスにリース制限を指定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが

	コマンドまたはアクション	目的
		終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

インターフェイスのリース制限の指定：例

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-interface 2400
 !
!
end
```

DHCP オプション 82 について

DHCP オプション 82 によって、DHCP サーバはクライアント デバイスのロケーションに基づいて IP アドレスを生成できます。 このオプションは、次のサブオプションを定義します。

- エージェント回線 ID のサブオプション：このサブ オプションは、DSLAM によって挿入され、DSLAM の加入者回線を識別します。
- エージェント リモート ID のサブオプション：このサブ オプションは、L2 接続されたトポロジの DSLAM または BNG によって挿入されます。 クライアント MAC アドレスですが、オーバーライドできます。 DHCP プロキシまたはリレーによって、クライアント MAC アドレスは、パケットが DHCP サーバに到達するまでに失われます。 これは、パケットがサーバに到達するときにクライアント MAC を維持するメカニズムです。

- VPN ID のサブオプション：このサブ オプションは、DHCP サーバに送信されるすべての DHCP 要求に VPN を伝えるためにリレー エージェントによって使用されます。また、DHCP サーバがリレー エージェントに返送する DHCP 応答の転送にも使用されます。
- サブネット選択のサブオプション：このサブ オプションは、IP アドレスからのサブネットの分離を可能にし、リレー エージェントとの通信に使用されます。 DHCP 処理では、DHCP クライアントが存在するサブネットと、リレーエージェントとの通信のためにサーバが使用する IP アドレスの両方が、ゲートウェイ アドレスによって指定されます。
- サーバ識別子オーバーライド サブオプション：このサブ オプション値は、通常のサーバ ID アドレスの代わりに DHCP サーバからの応答パケット内にコピーされます。 このサブ オプションには、クライアントからアクセス可能なリレー エージェントの IP アドレスである、着信インターフェイスの IP アドレスが含まれます。 この情報を使用して、DHCP クライアントは、リレー エージェントにすべての更新パケットとリリース パケットを送信し、これによって、元の DHCP サーバに VPN サブオプションのすべてを順番に追加し、更新パケットとリリース パケットを転送します。

（注） VPN ID、サブネット選択、およびサーバ識別子オーバーライド サブオプションは、MPLS VPN をサポートするための DHCP リレー/プロキシで使用されます。

オプション 82 のリレー情報のカプセル化

2 つのリレー エージェントが DHCP クライアントと DHCP サーバとの間でメッセージをリレーしているとき、デフォルトでは、2 番めのリレー エージェント（サーバに近いほう）が、当初の Option 82 情報を自身の Option 82 で置き換えます。 1 番めのリレー エージェントからのリモート ID および回線 ID 情報は失われます。 導入シナリオによっては、2 番めのリレー エージェントからの Option 82 だけでなく、1 番めのリレー エージェントからの初期の Option 82 も保持しておく必要がある場合があります。

DHCP オプション 82 のリレー情報のカプセル化機能を使用すると、独自のオプション 82 情報も追加するように設定してある場合、2 番目のリレー エージェントが 1 番目のリレー エージェントから受信したメッセージにオプション 82 情報をカプセル化できます。 この設定によって、DHCP サーバは両方のリレー エージェントからオプション 82 情報を使用できます。

DHCPv6 の概要

IPv6 のダイナミック ホスト コンフィギュレーション プロトコル（DHCPv6）によって、DHCP サーバは IPv6 ネットワーク アドレスなどの設定パラメータを IPv6 ノードに渡すことができます。ステートフルアドレス設定を使用して、要求クライアントに再利用可能なネットワーク アドレスを自動的に割り当てることができます。アドレスおよびプレフィックス割り当てとともに、DHCPv6 は、DNS アドレス、DNS ドメイン名、ネットワークの IPv6 ノードに対する AFTR アドレスなどの他の設定パラメータを割り当てることによって、柔軟な追加設定も可能にします。

基本的な DHCPv6 クライアント サーバの概念は、DHCP for IPv4（DHCPv4）の使用に似ています。クライアントが設定パラメータを受信する場合、接続されたローカル ネットワークで要求が送信され、使用可能な DHCPv6 サーバが検出されます。 DHCPv6 は、IPv6 アドレスまたはプレフィックス、ネーム サーバ、および DHCP for IPv4 のものとよく似た他の設定情報を割り当てますが、DHCPv4 と DHCPv6 には特定の大きな違いがあります。 たとえば、DHCPv4 とは異なり、DHCPv6 でのアドレス割り当てはメッセージ オプションを使用して処理され、DHCPv6 クライアントは 1 つの要求で複数のアドレスとプレフィックスを要求でき、DHCPv6 はアドレスおよびプレフィックスに異なるリース期間を要求できます。 これらの DHCPv6 の重要な利点により、DHCPv6 はアドレス割り当ての優先プロトコルになります。

IPv6 ホストは、ステートレス アドレス自動設定（SLAAC）、つまり、ローカルの情報とルータがアドバタイズした情報の組み合わせを使用してホストが独自のアドレスを生成するモデルを使用します。

DHCPv6 は、RFC 3315 で IETF によって標準化されています。 この DHCPv6 プロトコルは、IPv6 ステートレス アドレス自動設定（RFC 4862）へのステートフル カウンターパートで、設定パラメータを取得するために個別または SLAAC と同時に使用できます。

（注） DHCPv6 を設定する前に、DHCPv6 を提供するインターフェイスで IPv6 をイネーブルにして、ネイバー探索（ND）をイネーブルにする必要があります。

ネイバー探索（ND）の詳細情報については、『*Cisco IOS XR IP Addresses and Services Configuration Guide*』の「Implementing Network Stack IPv4 and IPv6」の項を参照してください。

制約事項

- DHCPv6 プロキシは、プロキシ プロファイルあたり最大 8 つの外部 DHCPv6 サーバをサポートします。
- バルク リース クエリーはサポートされません。
- DHCPv6 サーバは、BNG の設定でのみサポートされます。

DHCPv6 サーバおよび DHCPv6 リレーまたはプロキシ

DHCPv6 サーバは、常にステートフルなアドレス割り当てを使用します。 有効な要求を受信すると、DHCPv6 サーバは、IPv6 アドレスやプレフィックス、およびドメイン名、要求元のクライアントへのドメイン ネーム サーバ（DNS）などのその他の設定属性を割り当てます。

DHCPv6 リレーまたはプロキシは、クライアントからサーバに DHCPv6 メッセージを転送します。DHCPv6 リレーは、ステートレスまたはステートフルのいずれかのアドレス割り当てを使用できます。 DHCPv6 ステートレス リレー エージェントは、仲介装置としての役割を果たし、クライアントとサーバ間で DHCPv6 メッセージを配信します。 リレーは、クライアント アドレスまたはリース期間などの情報を保存または追跡しません。 DHCPv6 リレーは、ステートレス リレーとしても知られています。 一方、DHCPv6 ステートフル リレー エージェントは、DHCP プロキシとしても知られており、クライアントからサーバに DHCPv6 メッセージを転送するだけでなく、クラ

イアントのアドレスとリース期間の追跡も行います。 したがって、DHCPv6 プロキシは、ステートフル リレーとしても知られています。 DHCPv6 は、スタンドアロン プロキシをサポートします。

DHCPv6 プロキシによって、Remote-ID および Interface-ID オプションを挿入できます。 DHCPv6 プロキシは、Remote-ID に加えて Interface-ID を使用して、クライアントへの応答を送信するインターフェイスを選択します。

DHCPv6 は、別のコンフィギュレーション モードでイネーブルにできます。 異なるコンフィギュレーション モードでの DHCPv6 の設定の詳細については、異なるコンフィギュレーション モードの DHCPv6 のイネーブル化, （172 ページ）を参照してください。 DHCPv6 パラメータの設定の詳細については、DHCPv6 パラメータの設定, （178 ページ）を参照してください。

異なるコンフィギュレーション モードの DHCPv6 のイネーブル化

グローバル、サーバ プロファイル、プロキシ プロファイル コンフィギュレーション モードおよびサーバ プロファイル クラスおよびプロキシ プロファイル クラス サブ コンフィギュレーション モードなどの異なるコンフィギュレーション モードの DHCPv6 を有効にするには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **class** *class-name*
5. **dns-server** *address*
6. **domain-name** *name*
7. **prefix-pool** *pool_name*
8. **address-pool** *pool_name*
9. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
10. **interface** *type interface-path-id* **server profile** *profile_name*
11. **profile** *proxy_profile_name* **proxy**
12. **link-address** *ipv6_address*
13. **class** *class-name*
14. **helper-address vrf** *vrf_name ipv6_address*
15. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
16. **interface** *type interface-path-id* **proxy profile** *profile_name*
17. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**dhcp ipv6** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	DHCP for IPv6 を設定し、DHCPv6 コンフィギュレーション モードを開始します。
ステップ 3	**profile** *server_profile_name* **server** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	DHCPv6 サーバ プロファイルを作成し、DHCPv6 サーバ プロファイル サブコンフィギュレーション モードを開始します。
ステップ 4	**class** *class-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# class server-green`	サーバ プロファイルのクラスを定義し、サーバ プロファイル クラス サブモードを開始します。
ステップ 5	**dns-server** *address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# dns-server 1111::1`	サーバ プロファイルの DNS サーバおよびそれに対応するアドレスを定義します。
ステップ 6	**domain-name** *name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# domain-name www.xyz.com`	サーバプロファイルのドメイン名を定義します。
ステップ 7	**prefix-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# prefix_pool p1`	サーバ プロファイルのプレフィックス プールを設定します。
ステップ 8	**address-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# address_pool p1`	サーバ プロファイルのアドレス プールを設定します。

ステップ 9	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッショ

	コマンドまたはアクション	目的
		ンが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。
ステップ 10	**interface** *type interface-path-id* **server profile** *profile_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 server profile my-server-profile`	IPv6 インターフェイスに DHCPv6 サーバ設定プロファイルを関連付けます。
ステップ 11	**profile** *proxy_profile_name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-proxy-profile proxy`	DHCPv6 プロファイル プロキシを作成し、DHCPv6 プロキシ サブコンフィギュレーション モードを開始します。
ステップ 12	**link-address** *ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# link-address 5:6::78`	リレー転送メッセージのリンクアドレス フィールドに入力する IPv6 アドレスを指定します。
ステップ 13	**class** *class-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# class proxy-red`	プロキシ プロファイルのクラスを定義し、プロキシ プロファイル クラス サブモードを開始します。
ステップ 14	**helper-address vrf** *vrf_name ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	プロキシへのヘルパー アドレスとして DHCPv6 アドレスを設定します。 （注） ヘルパー アドレスは、プロキシ プロファイルおよびプロキシ プロファイル クラス サブモードでのみ設定できます。

	コマンドまたはアクション	目的

ステップ 15	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 16	**interface** *type interface-path-id* **proxy profile** *profile_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface BundleEther100.1 proxy profile my-proxy-profile`	IPv6 インターフェイスに DHCPv6 プロキシ設定プロファイルを関連付けます。
ステップ 17	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存さ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	れ、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

異なるコンフィギュレーション モードの DHCPv6 のイネーブル化：例

```
configure
dhcp ipv6
profile my-server-profile server
link-address 5:6::78
class server-green
dns-server 1111::1
domain-name www.cisco.com
prefix-pool POOL_P6_2
address-pool POOL_A6_1

end
!!
configure
dhcp ipv6
interface GigabitEthernet 0/2/0/0 server profile my-server-profile
profile my-proxy-profile proxy
link-address 5:6::78
class proxy-red
helper-address 5661:11
end
!!
configure
dhcp ipv6
interface GigabitEthernet 0/2/0/0 proxy profile my-proxy-profile
end
!!
```

DHCPv6 パラメータの設定

アドレス プール名、プレフィックス プール名、DNS サーバ、ドメイン名、リース期間、およびヘルパー アドレスなどの DHCPv6 パラメータを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **dns-server** *ipv6_address*
5. **domain-name** *domain_name*
6. **lease**
7. **helper-address vrf** *vrf_name ipv6_address*
8. **prefix-pool** *prefix-pool-name*
9. **address-pool** *address-pool-name*
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv6** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	DHCP for IPv6 を設定し、DHCPv6 コンフィギュレーション モードを開始します。
ステップ 3	**profile** *server_profile_name* **server** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	DHCPv6 サーバ プロファイルを設定し、DHCPv6 サーバ プロファイル サブコンフィギュレーション モードを開始します。
ステップ 4	**dns-server** *ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# dns-server 1:1:1::1`	DHCPv6 サーバ プロファイルの DNS サーバを設定します。

	コマンドまたはアクション	目的
		（注） DNS サーバ名は、クラス モードで定義されます。同じパラメータがプロファイルモードでも定義されている場合、クラスモードで定義された値が優先されます。
ステップ 5	**domain-name** *domain_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# domain-name my.domain.name`	DHCPv6 サーバ プロファイルの DNS ドメイン名を設定します。 （注） DNS サーバ名は、クラス モードで定義されます。同じパラメータがプロファイルモードでも定義されている場合、クラスモードで定義された値が優先されます。
ステップ 6	**lease** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# lease 1 6 0`	1 日、6 時間、および 0 分間のリース期間を設定します。
ステップ 7	**helper-address vrf** *vrf_name ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	プロキシへのヘルパー アドレスとして DHCPv6 アドレスを設定します。 （注） ヘルパー アドレスは、プロキシ プロファイルおよびプロキシプロファイル クラス サブモードでのみ設定できます。
ステップ 8	**prefix-pool** *prefix-pool-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile-class)# prefix-pool my-server-delegated-prefix-pool`	DHCPv6 サーバ プロファイル クラス サブモードでプレフィックス プールを設定します。
ステップ 9	**address-pool** *address-pool-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile-class)# address-pool my-server-address-pool`	DHCPv6 サーバ プロファイル クラス サブモードでアドレス プールを設定します。
ステップ 10	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

DHCPv6 パラメータの設定：例

```
configure
dhcp ipv6
profile my-server-profile server
dns-server 1:1:1::1
domain-name my.domain.name
lease 1 6 0
class class1
prefix-pool my-server-delegated-prefix-pool
address-pool my-server-address-pool
end
!!
```

DHCPv6 機能

DHCPv6 は、中央サーバからホスト IP アドレスを動的に割り当てるために、LAN 環境で広く使用されます。 このアドレスの動的な割り当ては、IP アドレス管理のオーバーヘッドを削減します。DHCPv6 は、限られた IP アドレス空間の節約にも役立ちます。 これは、IP アドレスを恒久的にホストに割り当てる必要がなくなり、ネットワークに接続されたホストだけが IP アドレスを使用するためです。

BNG でサポートされる DHCPv6 機能は、次のとおりです。

DHCPv6 のハイ アベイラビリティ サポート

DHCPv6 のハイ アベイラビリティ サポートは、次のとおりです。

ラインカードの活性挿抜

ラインカードの活性挿抜（OIR）によって、システムの動作に影響を与えることなく、欠陥のある部分を置き換えることができます。 カードが挿入された時点で、カードでは電源が使用可能で、自身で初期化を行って動作を開始します。

（注） DHCPv6 バインディングは、ラインカード OIR に影響されません。

チェック ポイントとシャドウ データベース

チェック ポイントとシャドウ データベースは、RSP でアクティブに保持され、すべてのラインカードからのすべてのバインディングのコピーが含まれます。 チェック ポイント データベースには、その範囲のインターフェイス上の加入者からのクライアントまたは加入者バインディングがあります。 アクティブな RSP のシャドウ データベースは、スタンバイのシャドウ データベースを更新します。

DHCPv6 ホット スタンバイ

DHCPv6 ホット スタンバイは、RSP でのみサポートされるプロセスです。 アクティブな RSP が応答を停止するたびに、スタンバイ RSP と即座に置き換えられます。 スタンバイ RSP は、アクティブになると処理を引き継ぎます。

DHCPv6 システム リロードの持続的なバインディング

DHCPv6 システム リロードの持続的なバインディング機能は、DHCPv6 サーバおよび DHCPv6 プロキシのシステム リロードによってバインディング テーブルの永続性を実現します。 これにより、DHCP サーバでのリロード イベントの発生に関係なく、DHCP クライアントは DHCP リースを維持できます。 DHCPv6 Proxy では、この永続性は、スタンドアロンおよび IPoE セッションの両方に対してサポートされます。 DHCPv6 サーバでは、永続性は IPoE セッションに対してのみサポートされます。

システム リロードの永続的なバインディング機能が DHCPv6 に対してイネーブルな場合、DHCP バインディングはルータのファイル システムに保存されます。 完全な永続ファイルの書き込みと増分ファイルの書き込みは、設定可能な間隔で DHCP バインディングを保存するためにファイル システムで発生します。

DHCP の設定とコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』の「*Implementing the Dynamic Host Configuration Protocol*」の章と『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*DHCP Commands*」の章を参照してください。

DHCPv6 プレフィックス委任

DHCPv6 のプレフィックス委任は、IPv6 のプレフィックスをクライアントに委任するメカニズムです。プレフィックス委任機能を使用して、リンク、サブネット、およびサイトアドレッシングの変更を管理できます。

インターネット サービス プロバイダー（ISP）は、カスタマーのネットワーク内で使用するために、カスタマーにプレフィックスを割り当てます。 プレフィックス委任は、DHCPv6 プレフィックス委任オプションを使用して、プロバイダー エッジ（PE）デバイスと宅内装置（CPE）の間で行われます。 ISP によってプレフィックスがカスタマーに委任されると、カスタマーはさらにプレフィックスをサブネット化してカスタマーのネットワーク内のリンクに割り当てます。

デフォルトでは、プレフィックス委任機能は常にイネーブルです。

IPv6 IPoE 加入者サポート

IPv6 加入者は、DHCPv6 プロトコルを使用して作成された IPv6 アドレスを送信します。 IPv6 加入者は、CPE デバイスで IPv6 を実行し、レイヤ 2 ネットワークまたはレイヤ 2 集約経由で BNG に接続されます。 IPv6 加入者は、BNG にまたはレイヤ 2 アグリゲータを介して直接接続されている場合にサポートされます。

IPv6 IPoE 加入者サポートをイネーブルにするには、DHCPv6 プロファイルを加入者インターフェイスで明示的に設定する必要があります。 詳細については、「IPv6 IPoE 加入者インターフェイスの設定, （182 ページ）」を参照してください。

FSOL の処理

DHCPv6 First Sign of Life（FSOL）処理は、IPoE セッションでのみサポートされます。 DHCPv6 は、IPoEセッションの確認および作成のために、FSOLパケットとしてクライアントからのSOLICITパケットを処理します。 IPoE セッションは、設定が存在し、加入者情報が正常に検証される限り、作成されます。

IPv6 IPoE 加入者インターフェイスの設定

IPoE 加入者インターフェイスを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *name* **ipv6** *pool_name*
3. **address-range** *first_ipv6_address last_ipv6_address*
4. **pool vrf** *name* **ipv6** *pool_name*
5. **prefix-length** *length*
6. **prefix-range** *first_ipv6_address last_ipv6_address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
8. **dhcp ipv6**
9. **interface** *type interface-path-id* **server profile** *profile_name*
10. **profile** *server_profile_name* **server**
11. **prefix-pool** *pool_name*
12. **address-pool** *pool_name*
13. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
14. **dhcp ipv6**
15. **interface** *type interface-path-id* **proxy profile** *profile_name*
16. **profile** *server_profile_name* **proxy**
17. **helper-address vrf** *vrf_name ipv6_address*
18. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
19. **dynamic-template type ipsubscriber** *dynamic_template_name*
20. **ipv6 enable**
21. **dhcpv6 address-pool** *pool_name*
22. **dhcpv6 delegated-prefix-pool** *pool_name*
23. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
24. **class-map type control subscriber match-all** *class-map_name*
25. **match protocol dhcpv6**

26. **end-class-map**
27. **policy-map type control subscriber** *class-map_name*
28. **event session-start match-first**
29. **class type control subscriber** *class_name* **do-all**
30. *sequence_number* **activate dynamic-template** *dynamic-template_name*
31. **end-policy-map**
32. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
33. **interface** *type interface-path-id*
34. **ipv4 address** *ipv4_address*
35. **ipv6 address** *ipv6_address*
36. **ipv6 enable**
37. **service-policy type control subscriber** *name*
38. **ipsubscriber ipv6 l2-connected**
39. **initiator dhcp**
40. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool vrf** *name* **ipv6** *pool_name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 pool1`	分散アドレス プール サービスを設定します。
ステップ 3	**address-range** *first_ipv6_address last_ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# address-range 2201:abcd:1234:2400:f800::1 2201:abcd:1234:2400:f800::fff`	アドレス範囲を設定します。

	コマンドまたはアクション	目的
ステップ 4	**pool vrf** *name* **ipv6** *pool_name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 pool2`	分散アドレス プール サービスを設定します。
ステップ 5	**prefix-length** *length* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 92`	使用するプレフィックス長を指定します。
ステップ 6	**prefix-range** *first_ipv6_address last_ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-range 3301:1ab7:2345:1200:f800:: 3301:1ab7:2345:1200:f800:fff0::`	割り当てのプレフィックス範囲を指定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 8	**dhcp ipv6** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	DHCP for IPv6 を設定し、DHCPv6 コンフィギュレーション モードを開始します。
ステップ 9	**interface** *type interface-path-id* **server profile** *profile_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 server profile foo`	IPv6 インターフェイスに DHCPv6 プロキシ設定プロファイルを関連付けます。
ステップ 10	**profile** *server_profile_name* **server** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile foo server`	DHCPv6 サーバプロファイルを作成し、DHCPv6 サーバプロファイル サブコンフィギュレーション モードを開始します。
ステップ 11	**prefix-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# prefix-pool pool2`	サーバプロファイルのプレフィックス プールを設定します。
ステップ 12	**address-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# address-pool pool1`	サーバプロファイルのアドレス プールを設定します。

ステップ 13	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続しま

	コマンドまたはアクション	目的
		す。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 14	**dhcp ipv6** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	DHCP for IPv6 を設定し、DHCPv6 コンフィギュレーション モードを開始します。
ステップ 15	**interface** *type interface-path-id* **proxy profile** *profile_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 proxy profile foo`	IPv6 インターフェイスに DHCPv6 プロキシ設定プロファイルを関連付けます。
ステップ 16	**profile** *server_profile_name* **proxy** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile foo proxy`	DHCPv6 サーバプロファイルを作成し、DHCPv6 サーバプロファイルサブコンフィギュレーション モードを開始します。
ステップ 17	**helper-address vrf** *vrf_name ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	プロキシへのヘルパー アドレスとして DHCPv6 アドレスを設定します。 （注） ヘルパー アドレスは、プロキシ プロファイルおよびプロキシプロファイル クラス サブモードでのみ設定できます。

ステップ 18	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 19	**dynamic-template type ipsubscriber** *dynamic_template_name* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ipsubscriber dhcpv6_temp`	ipsubscriber タイプの動的なテンプレートを設定し、動的なテンプレート タイプのコンフィギュレーション モードを開始します。
ステップ 20	**ipv6 enable** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 enable`	インターフェイス上で IPv6 をイネーブルにします。
ステップ 21	**dhcpv6 address-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool pool3`	DHCPv6 アドレス プールを設定します。
ステップ 22	**dhcpv6 delegated-prefix-pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 delegated-prefix-pool pool4`	DHCPv6 の委任されたプレフィックス プールを設定します。
ステップ 23	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 24	**class-map type control subscriber match-all** *class-map_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all dhcpv6_class`	match-any 基準でクラス マップ コントロール加入者を設定します。
ステップ 25	**match protocol dhcpv6** 例： `RP/0/RSP0/CPU0:router(config-cmap)# match protocol dhcpv6`	前述の手順で設定されたクラスの一致基準を設定します。
ステップ 26	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	最後のクラス マップを設定します。
ステップ 27	**policy-map type control subscriber** *class-map_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber dhcpv6-policy`	加入者コントロール ポリシーマップを設定します。

	コマンドまたはアクション	目的
ステップ 28	**event session-start match-first** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first`	match-first 基準でポリシー イベントを設定します。
ステップ 29	**class type control subscriber** *class_name* **do-all** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber dhcpv6_class do-all`	match-any 基準でクラス マップ コントロール加入者を設定します。
ステップ 30	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 20 activate dynamic-template dhcpv6_temp`	動的なテンプレートに関連するアクションをアクティブ化します。
ステップ 31	**end-policy-map** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	最後のポリシー マップを設定します。

ステップ 32	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 33	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.1`	インターフェイスを設定して、インターフェイス コンフィギュレーション モードを開始します。
ステップ 34	**ipv4 address** *ipv4_address* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 11.11.11.2 255.255.255.0`	インターフェイスに IPv4 アドレスを設定します。
ステップ 35	**ipv6 address** *ipv6_address* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 11:11:11::2/64`	インターフェイスに IPv6 アドレスを設定します。
ステップ 36	**ipv6 enable** 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	インターフェイス上で IPv6 をイネーブルにします。
ステップ 37	**service-policy type control subscriber** *name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber dhcpv6_policy`	インターフェイスに加入者制御サービス ポリシーを関連付けます。
ステップ 38	**ipsubscriber ipv6 l2-connected** 例： `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`	L2 接続された IPv6 加入者をイネーブルにします。
ステップ 39	**initiator dhcp** 例： `RP/0/RSP0/CPU0:router(config-if-ipsub-ipv6-l2conn)# initiator dhcp`	IPv6 加入者の発信側を設定します。
ステップ 40	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		`before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPv6 IPoE 加入者インターフェイスの設定：例

```
configure
pool vrf default ipv6 pool1
 address-range 2201:abcd:1234:2400:f800::1 2201:abcd:1234:2400:f800::fff

pool vrf default ipv6 pool2
prefix-length 92
prefix-range 3301:1ab7:2345:1200:f800:: 3301:1ab7:2345:1200:f800:fff0::

dhcp ipv6
 interface GigabitEthernet0/3/0/0 server profile foo
 profile foo server
  prefix-pool pool2
  address-pool pool1
!
!
end

configure
dhcp ipv6
 interface GigabitEthernet0/3/0/0 proxy profile foo
 profile foo proxy
  helper address <v6 address of the server
!
!
```

```
dynamic-template type ipsubscriber dhcpv6_temp
  ipv6 enable
  dhcpv6 address-pool pool3
  dhcpv6 delegated-prefix-pool pool4
 !
!
!
class-map type control subscriber match-all dhcpv6_class
 match protocol dhcpv6
 end-class-map
!
policy-map type control subscriber dhcpv6_policy
 event session-start match-first
  class type control subscriber dhcpv6_class do-all
   20 activate dynamic-template dhcpv6_temp
!
!
end

configure
interface GigabitEthernet0/3/0/0
 ipv4 address 11.11.11.2 255.255.255.0
 ipv6 address 11:11:11::2/64
 ipv6 enable
 service-policy type control subscriber dhcpv6_policy
 ipsubscriber ipv6 l2-connected
  initiator dhcp
!
!
end
end
```

IPv6 PPPoE 加入者サポート

PPPoE 加入者インターフェイスは、認証およびアドレス割り当てに使用される加入者で PPP リンクを確立します。 DHCPv6 サーバは、PPPoE 加入者にアドレスまたはプレフィックスを割り当てます。 PPPoE 加入者インターフェイスは動的に作成されるため、DHCPv6 プロファイルは、単一の PPPoE インターフェイスのみでなく、ルータで作成されるすべての PPPoE インターフェイスに適用されます。

PPPoE 加入者サポートをイネーブルにするには、DHCPv6 プロファイルをグローバルに設定するか、すべての PPPoE インターフェイスで設定する必要があります。 詳細については、「IPv6 PPPoE 加入者インターフェイスの設定, （193 ページ）」を参照してください。

IPv6 PPPoE 加入者インターフェイスの設定

PPPoE 加入者インターフェイスを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template type ppp** *dynamic_template_name*
3. **ppp authentication chap**
4. **ppp ipcp peer-address pool** *pool_name*
5. **ipv4 unnumbered** *interface-type interface-path-id*
6. **ipv6 enable**
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
8. **class-map type control subscriber match-any** *class-map_name*
9. **match protocol ppp**
10. **end-class-map**
11. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
12. **class-map type control subscriber match-all** *class-map_name*
13. **match protocol dhcpv6**
14. **end-class-map**
15. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
16. **policy-map type control subscriber** *policy_name*
17. **event session-start match-first**
18. **class type control subscriber** *name* **do-all**
19. *sequence_number* **activate dynamic-template** *dynamic-template_name*
20. **end-policy-map**
21. **policy-map type control subscriber** *policy_name*
22. **event session-start match-all**
23. **class type control subscriber** *name* **do-all**
24. *sequence_number* **activate dynamic-template** *dynamic-template_name*
25. **end-policy-map**
26. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

27. **interface** *type interface-path-id*
28. **description** *LINE*
29. **ipv6 enable**
30. **service-policy type control subscriber** *name*
31. **encapsulation dot1q 801**
32. **ipsubscriber ipv6 l2-connected**
33. **initiator dhcp**
34. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template type ppp** *dynamic_template_name* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp ppp_pta_template`	PPP タイプの動的なテンプレートを設定し、動的なテンプレート タイプのコンフィギュレーション モードを開始します。
ステップ 3	**ppp authentication chap** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication chap`	チャレンジ ハンドシェイク認証プロトコル（chap）を設定し、PPP リンク認証方式を設定します。
ステップ 4	**ppp ipcp peer-address pool** *pool_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp ipcp peer-address pool p1`	IPCP ネゴシエーション オプションを設定し、ピア アドレス プールのピア アドレス設定オプションを設定します。
ステップ 5	**ipv4 unnumbered** *interface-type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 unnumbered Loopback 1`	インターフェイスの明示的なアドレスを使用せずに IPv4 処理をイネーブルにします。
ステップ 6	**ipv6 enable** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 enable`	インターフェイス上で IPv6 をイネーブルにします。

	コマンドまたはアクション	目的

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 8	**class-map type control subscriber match-any** *class-map_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-any pta_class`	match-any 基準でクラス マップ コントロール加入者を設定します。
ステップ 9	**match protocol ppp** 例： `RP/0/RSP0/CPU0:router(config-cmap)# match protocol ppp`	前述の手順で設定されたクラスの一致基準を設定します。
ステップ 10	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	最後のクラス マップを設定します。

	コマンドまたはアクション	目的

ステップ 11	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 12	**class-map type control subscriber match-all** *class-map_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all ipoe_test`	match-all 基準でクラス マップ コントロール加入者を設定します。
ステップ 13	**match protocol dhcpv6** 例： `RP/0/RSP0/CPU0:router(config-cmap)# match protocol dhcpv6`	前述の手順で設定されたクラスの一致基準を設定します。
ステップ 14	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	最後のクラス マップを設定します。

	コマンドまたはアクション	目的

ステップ 15	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 16	**policy-map type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber policy1`	加入者コントロール ポリシーマップを設定します。
ステップ 17	**event session-start match-first** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first`	match-first 基準でポリシー イベントを設定します。
ステップ 18	**class type control subscriber** *name* **do-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type control subscriber ipoe_test1 do-all`	match-first 基準でポリシー イベントを設定します。

	コマンドまたはアクション	目的
ステップ 19	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 24 activate dynamic-template v6_test1`	動的なテンプレートに関連するアクションをアクティブ化します。
ステップ 20	**end-policy-map** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	最後のポリシー マップを設定します。
ステップ 21	**policy-map type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber policy1`	加入者コントロール ポリシーマップを設定します。
ステップ 22	**event session-start match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	match-all 基準でポリシー イベントを設定します。
ステップ 23	**class type control subscriber** *name* **do-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type control subscriber pta_class do-all`	match-first 基準でポリシー イベントを設定します。
ステップ 24	*sequence_number* **activate dynamic-template** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ppp_pta_template`	動的なテンプレートに関連するアクションをアクティブ化します。
ステップ 25	**end-policy-map** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	最後のポリシー マップを設定します。
ステップ 26	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存さ

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	れ、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 27	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface BundleEther1.1`	インターフェイスを設定して、インターフェイス コンフィギュレーション モードを開始します。
ステップ 28	**description** *LINE* 例： `RP/0/RSP0/CPU0:router(config-if)# description IPoE`	上記の設定済みインターフェイスの説明を設定します。
ステップ 29	**ipv6 enable** 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	インターフェイス上でIPv6をイネーブルにします。
ステップ 30	**service-policy type control subscriber** *name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber ipoe1`	インターフェイスに加入者制御サービス ポリシーを関連付けます。
ステップ 31	**encapsulation dot1q 801** 例： `RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 801`	カプセル化された802.1Q VLAN設定をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 32	**ipsubscriber ipv6 l2-connected** 例： `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`	L2 接続された IPv6 加入者をイネーブルにします。
ステップ 33	**initiator dhcp** 例： `RP/0/RSP0/CPU0:router(config-if-ipsub-ipv6-l2conn)# initiator dhcp`	IPv6 加入者の発信側を設定します。

ステップ 34	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPv6 PPPoE 加入者インターフェイスの設定：例

```
configure
dynamic-template
type ppp PPP_PTA_TEMPLATE
ppp authentication chap
ppp ipcp peer-address pool ADDRESS_POOL
ipv4 unnumbered Loopback0
ipv6 enable
!
type ipsubscriber v6_test1
ipv6 enable
!
!
class-map type control subscriber match-any PTA_CLASS
match protocol ppp
end-class-map
!
class-map type control subscriber match-all ipoe_test1
match protocol dhcpv6
end-class-map
!
policy-map type control subscriber ipoe1
event session-start match-first
class type control subscriber ipoe_test1 do-all
24 activate dynamic-template v6_test1
!
!
end-policy-map
!
policy-map type control subscriber POLICY1
event session-start match-all
class type control subscriber PTA_CLASS do-all
1 activate dynamic-template PPP_PTA_TEMPLATE
!
!
end-policy-map
!
interface Bundle-Ether2.801
description IPoE
ipv6 enable
service-policy type control subscriber ipoe1
encapsulation dot1q 801
ipsubscriber ipv6 l2-connected
initiator dhcp
```

あいまいな VLAN サポート

あいまいな VLAN は、VLAN ID の範囲またはグループで設定されます。 あいまいな VLAN に作成された加入者セッションは、ポリシーマップ、VRF、QoS、および ACL などすべての通常設定をサポートする通常の VLAN 上の加入者と同じです。 複数の加入者は、一意の MAC アドレスが含まれている限り、特定の VLAN ID で作成できます。 あいまいな VLAN は、複数のアクセス インターフェイスを設定する必要性を減らすことによって、スケーラビリティを向上します。

DHCPv6 サポートをイネーブルにするには、あいまいな VLAN をバンドル インターフェイスの上でアンナンバードにします。

（注） あいまいな VLAN は、通常の VLAN とまったく同じ方法で名付けられます。 あいまいな VLAN は、l2transport インターフェイスで許可される EFP 範囲と対照的に、レイヤ 3 インターフェイスと見なされます。

DHCPv6 サーバがあいまいな VLAN インターフェイスで SOLICIT メッセージを受信すると、VLAN ID が受信パケットから抽出され、クライアントの関連情報とともに加入者の認証に使用されます。

インターフェイス設定があいまいから非あいまいに変更されたり、その逆になったり、またはあいまいな VLAN 範囲が変更されると、あいまいな VLAN に対するすべての既存のクライアントバインディングは削除されます。

あいまいな VLAN の設定の詳細については、あいまいな VLAN の設定, （203 ページ）を参照してください。

あいまいな VLAN の設定

あいまいな VLAN を設定するには、次の作業を実行します。

（注） あいまいな VLAN に必要な DHCP 固有の設定はありません。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のカプセル化のいずれかを使用して、カプセル化されたあいまいな VLAN を設定します。
   - **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* }
   - **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* }
   - **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* }
   - **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
4. **ipv4 | ipv6address** *source-ip-address destination-ip-address*
5. **service-policy type control subscriber** *policy_name*
6. **ipsubscriber** { **ipv4**|**ipv6** } **l2-connected**
7. **initiator dhcp**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.12`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

ステップ 3	次のカプセル化のいずれかを使用して、カプセル化されたあいまいな VLAN を設定します。
	• **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* }
	• **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* }
	• **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* }
	• **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 14 second-dot1q 100-200`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any second-dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1ad 14 dot1q 100,200,300-400`
		IEEE 802.1Q VLAN を設定します。
		vlan-range は、例に示すように、カンマ区切りまたはハイフン区切り形式、または両方の組み合わせで指定できます。

ステップ 4	**ipv4 | ipv6address** *source-ip-address destination-ip-address*
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 2.1.12.1 255.255.255.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv6 address 1:2:3::4 128`
		IPv4 または IPv6 プロトコル アドレスを設定します。

ステップ 5	**service-policy type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	指定された PL1 の policy_name でポリシーマップが前に定義された、アクセスインターフェイスにポリシーマップを適用します。

	コマンドまたはアクション	目的

ステップ 6	**ipsubscriber** { **ipv4**|**ipv6** } **l2-connected**
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv4 l2-connected`
	`RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`
		L2 接続された IPv4 または IPv6 IP 加入者をイネーブルにします。

ステップ 7	**initiator dhcp** 例： `RP/0/RSP0/CPU0:router(config-if)# initiator dhcp`	IP 加入者の発信側 DHCP をイネーブルにします。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

あいまいな VLAN の設定：例

```
configure
interface Bundle-Ether100.12
encapsulation ambiguous dot1q 14 second-dot1q any
```

```
ipv4 address 2.1.12.1 255.255.255.0
service-policy type control subscriber PL1
ipsubscriber ipv4 l2-connected
initiator dhcp
!
!
end
```

DHCPv6 アドレスまたはプレフィックス プール

アドレスまたはプレフィックス プールは、委任ルータがアドレスを割り当てるか、要求側ルータにプレフィックスを委任する使用可能なアドレスまたはプレフィックスプールを表します。分散アドレス プール サービス（DAPS）は、DHCPv6 のアドレスまたはプレフィックス プールを管理および維持します。

DHCPv6 プレフィックス委任には、プレフィックスを選択し、要求側ルータに一時的に委任している委任ルータが含まれます。委任ルータは、アドレスを割り当てるか、アドレスプールまたはプレフィックス プールから要求側ルータにプレフィックスを委任します。

DHCPv6 アドレスまたはプレフィックス プールの設定の詳細については、IPv6 アドレスまたはプレフィックス プール名の設定, （206 ページ）を参照してください。

IPv6 アドレスまたはプレフィックス プール名の設定

動的テンプレート コンフィギュレーション モードで IPv6 アドレスまたはプレフィックス プール名を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type ipsubscriber** *dynamic-template_name*
4. **dhcpv6 delegated-prefix-pool** *pool-name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
6. **type ppp** *dynamic-template_name*
7. **dhcpv6 address-pool** *pool-name*
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
9. **type ipsubscriber** *dynamic-template_name*
10. **dhcpv6 address-pool** *pool-name*
11. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**
12. **ipv6 nd framed-prefix-pool** *pool-name*
13. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的なテンプレートの設定をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**type ipsubscriber** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipv6-sub-template`	ipsubscriber タイプの動的なテンプレートを設定し、動的なテンプレートタイプのコンフィギュレーション モードを開始します。
ステップ 4	**dhcpv6 delegated-prefix-pool** *pool-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 delegated-prefix-pool mypool`	プレフィックス委任プールに IPv6 加入者の動的なテンプレートを設定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 6	**type ppp** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp ipv6-sub-template`	PPP タイプの動的なテンプレートを設定します。

	コマンドまたはアクション	目的
ステップ 7	**dhcpv6 address-pool** *pool-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool my-pppoe-addr-pool`	PPPoE 加入者の IPv6 アドレス プールを設定します。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 9	**type ipsubscriber** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber my-ipv6-template`	ipsubscriber タイプの動的なテンプレートを設定し、動的なテンプレート タイプのコンフィギュレーション モードを開始します。
ステップ 10	**dhcpv6 address-pool** *pool-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool my-ipsub-addr-pool`	IPoE 加入者の IPv6 アドレス プールを設定します。
ステップ 11	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 12	**ipv6 nd framed-prefix-pool** *pool-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# framed-prefix-pool my-slaac-pool`	SLAAC のみが使用するようにプレフィックス プールを設定します。

ステップ 13	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが

	コマンドまたはアクション	目的
		EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPv6 アドレスまたはプレフィックス プール名の設定：例

```
configure
dynamic-template
type ipsubscriber ipv6-sub-template
dhcpv6 delegated-prefix-pool mypool
end
dynamic-template
type ppp ipv6-sub-template
dhcpv6 address-pool my-pppoe-addr-pool
!
type ipsubscriber my-ipv6-template
dhcpv6 address-pool my-ipsub-addr-pool
!!
ipv6 nd framed-prefix-pool my-slaac-pool
end
!!
```

DHCPv6 Dual-Stack Lite サポート

Dual-Stack Lite（DS-Lite）は、ホストとルータの両方で、IPv4 と IPv6 の両方のインターネット プロトコルに完全なサポートを提供する手法です。 Dual-Stack Lite によって、ブロードバンド サービス プロバイダーは、IP in IP（IPv4- in-IPv6）およびネットワーク アドレス変換（NAT）の 2 種類のテクノロジーを統合することでカスタマーと IPv4 アドレスを共有できます。

DS-Lite 機能には、基本的なブリッジング ブロードバンド（B4）とアドレス ファミリ遷移ルータ（AFTR）の 2 つのコンポーネントが含まれます。

B4 要素は、直接接続されたデバイスまたはアドレス ファミリ遷移ルータ（AFTR）にトンネルを作成する CPE のいずれかの Dual-Stack 対応ノードに実装された機能です。 一方、AFTR 要素は、IPv4-in-IPv6 トンネル エンドポイントおよび同じノードに実装された IPv4-IPv4 NAT の組み合わせです。 DS-Lite B4 要素は、対応する AFTR ロケーションの IPv6 アドレスを検出する DHCPv6 オプションを使用します。

DS-Lite の AFTR の設定の詳細については、DS-Lite の AFTR 完全修飾ドメイン名の設定, （212 ページ）を参照してください。

DS-Lite の AFTR 完全修飾ドメイン名の設定

DS-Lite の AFTR 完全修飾ドメイン名を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **aftr-name** *aftr_name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dhcp ipv6** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	DHCP for IPv6 を設定し、DHCPv6 コンフィギュレーション モードを開始します。
ステップ 3	**profile** *server_profile_name* **server** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	DHCPv6 サーバ プロファイルを設定し、DHCPv6 サーバ プロファイル サブコンフィギュレーション モードを開始します。
ステップ 4	**aftr-name** *aftr_name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# aftr-name aftr-server.example.com`	サーバ プロファイル モードで、DS-Lite サポートの AFTR 完全修飾ドメイン名オプションを設定します。
ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

DS-Lite の AFTR 完全修飾ドメイン名の設定：例

```
configure
dhcp ipv6
profile my-server-profile server
aftr-name aftr-server.example.com
end
!!
```

DHCPv6 の VRF 認識

VRF 認識は、同じ IP アドレスが異なる VPN のクライアントに割り当てられている場合に、異なる VPN で複数のクライアントをサポートする DHCPv6 サーバまたはプロキシの機能です。 VRF の IPv6 アドレスは、別の VRF の IPv6 アドレスから独立しています。 複数の VRF で同じプレフィックス/アドレスを持つことは必須ではありません。

動的なテンプレートでの VRF の定義の詳細については、動的なテンプレートでの VRF の定義, （214 ページ）を参照してください。

動的なテンプレートでの VRF の定義

動的なテンプレートで VRF を定義するには、次の作業を実行します。 VRF の IPv6 アドレスは、別の VRF の IPv6 アドレスから独立しています。 複数の VRF で同じプレフィックスまたはアドレスを持つことは必須ではありません。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type ipsubscriber** *dynamic-template_name*
4. **vrf** *vrf_name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的なテンプレートの設定をイネーブルにします。
ステップ 3	**type ipsubscriber** *dynamic-template_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipv6-sub-template`	ipsubscriber タイプの動的なテンプレートを設定し、動的なテンプレート タイプのコンフィギュレーション モードを開始します。
ステップ 4	**vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# vrf vrf1`	インターフェイスが動作する VRF を設定します。
ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

動的なテンプレートでの VRF の定義：例

```
configure
dynamic-template
type ipsubscriber ipv6-sub-template
vrf vrf1
end
!!
```

加入者インターフェイスでのパケット処理

この項では、加入者インターフェイスが特定の特殊ケースでどのようにサポートされるかについて説明します。 これらの特殊ケースには、L3 転送されるインターフェイスが含まれます。 結果として、このサポートは、PPP over Ethernet PPP Termination and Aggregation（PPPoE PTA）と IPoE セッションにのみ適用できます。

ほとんどの加入者データ パケットは、ネットワーク処理装置（NPU）によって直接転送されます。 NPUが完全にデータ パケットを処理しない特定の特殊ケースがあります。 これらの特殊ケースはCPUで処理され、この目的のために作成された内部インターフェイスを通過します。 この内部インターフェイスの名前は、加入者インターフェイスまたはSINT です。 SINT は、加入者インターフェイスでパントされるすべてのパケットで使用される集約インターフェイスです。 各ノードに 1 つの SINT があります。 BNG パッケージをインストールすると、デフォルトで SINT が作成されます。 SINTインターフェイスは、加入者インターフェイスでのパケットのパントインジェクトに必要です。

これらの特殊ケースは、IPoE および PPPoE の両方 PTA についてサポートされます。

（注） これらの特殊ケースは、L2 サービスであるため、PPPoE L2TP には適用されません。

- 加入者に対する ping

  BNG は、IPoE および PPPoE PTA 加入者インターフェイスの両方からの ping 要求の受信を許可します。これは、他の非 BNG インターフェイス タイプと同じです。 同様に、BNG は、IPoE および PPPoE PTA 加入者インターフェイスの両方への ping 要求の送信も許可します。次の内容が含まれています。

  - 加入者の MTU サイズを超える長さを含むさまざまな ping パケット長
  - デフォルトおよびプライベート VRF の加入者
  - タイプ オブ サービス、DF セット、および冗長などのさまざまな ping オプション

  BNG は、IPv4 および IPv6 加入者の両方からの ping 要求の受信もサポートします。

  （注） 過剰なパント フロー トラップ機能は、加入者インターフェイスとの間で高レートの ping を送信すると、ディセーブルになります。

- オプションの処理

  BNG は IP オプションの処理をサポートします。これは、非 BNG インターフェイス タイプと同じです。 これらは NPU から CPU にパントされます。 これらは SINT インターフェイスを経由して、適切なアプリケーションで処理されます。

- traceroute、PMTU ディスカバリ、ICMP 到達不能のサポート

  - BNG は、転送できない PPPoE または IP 加入者インターフェイスとやり取りされるパケットの ICMP の送信をサポートします。 この機能は、他の非 BNG 加入者インターフェイスに似ています。
  - BNG は、パケットが加入者インターフェイスに送信されるときには、BNG が ICMP を送信する PMTU をサポートしますが、パケットは加入者の MTU を超過し、DF ビットが設定されます。

- BNG は、加入者インターフェイスとやり取りされるパケット（出力 ACL と入力 ACL）が ACL によって拒否されるときに ICMP の送信をサポートします。 ACL が拒否とロギングの両方を実行するように設定されている場合、パケットはドロップされますが、ICMP は生成されません。
- BNG は、パケットの存続可能時間（TTL）を超過すると ICMP の送信をイネーブルにする traceroute 機能をサポートします。
- BNG は、IPv4 と IPv6 加入者の両方に対して traceroute 機能をサポートします。

- フラグメンテーション

  BNG は、発信 MTU を超過する PPPoE または IP 加入者インターフェイスに送信されるパケットのフラグメンテーションをサポートします。

注意 フラグメンテーションを必要とするすべてのパケットは、NPU ごとに最大 1000 pps にポリシングされます。

（注） フラグメント パケットは、加入者の出力 QoS アクションをスキップします。

制約事項

次の制約事項が、加入者インターフェイスの実装に適用されます。

- ACL ロギング中、パケットは CPU にパントされ、BNG インターフェイスは SINT インターフェイスに誘導されます。 システムが BNG インターフェイスで ACL ロギングをサポートしないため、SINT インターフェイスはこれらのログ パケットをドロップします。
- IPv6 の ping および traceroute 機能は、CPE および BNG ルータ両方のグローバル アドレスを使用する必要があります。 リンク ローカル アドレスを使用する IPv6 の ping および traceroute 機能は、すべての場合で機能しません。
- 加入者 ACL でのロギングは、サポートされていません。

IPv6 ネイバー探索

IPv6 ネイバー探索プロセスでは、インターネット制御メッセージ プロトコル（ICMP）メッセージおよび送信要求ノード マルチキャスト アドレスを使用して、同じネットワーク（ローカル リンク）上のネイバーのリンク層アドレスを判断し、ネイバーに到達可能かどうかを確認し、隣接ルータを追跡します。

ネイバー探索用の IPv6 スタティック キャッシュ エントリ機能により、IPv6 ネイバー キャッシュ内にスタティック エントリを作成できます。 スタティック ルーティングでは、管理者が手動でテーブルに、各デバイスの各インターフェイスの IPv6 アドレス、サブネット マスク、ゲートウェイ、および対応するメディア アクセス コントロール（MAC）のアドレスを入力する必要があり

ます。スタティック ルーティングによって、より詳細な制御が可能になりますが、テーブルの保守作業が増えます。ルートが追加または変更されるたびにテーブルを更新する必要があります。

ネイバー探索の異なるメッセージ タイプは、次のとおりです。

- IPv6 ネイバー送信要求メッセージ：ICMP パケット ヘッダーのタイプ フィールドの値 135 は、ネイバー送信要求メッセージを識別します。ネイバー送信要求メッセージは、ノードが同じローカル リンク上の別のノードのリンク層アドレスを決定するときに、ローカル リンク上で送信されます
- IPv6 ルータ アドバタイズメント メッセージ：ICMP パケット ヘッダーのタイプ フィールドの値が 134 であるルータ アドバタイズメント（RA）メッセージは、定期的に IPv6 デバイスの設定された各インターフェイスに送信されます。
- IPv6 ネイバー リダイレクト メッセージ：ICMP パケット ヘッダーのタイプ フィールドの値 137 は、IPv6 ネイバー リダイレクト メッセージを識別します。デバイスは、ネイバー リダイレクト メッセージを送信して、宛先へのパス上のより適切なファーストホップ ノードをホストに通知します

BNG では、IPv6 ネイバー探索は IPoE および PPPoE セッションの両方をサポートします。IPv6 ネイバー探索は、PPPoE 加入者にプレフィックスを割り当てる際に使用するステートレス アドレス自動設定（SLAAC）を提供します。

その他の関連資料

ここでは、PPP、PPPoE、L2TP、および DHCP の実装に関連する参考資料を示します。

RFC

標準/RFC：PPP	タイトル
RFC-1332	『The PPP Internet Protocol Control Protocol（IPCP）』
RFC-1570	『PPP LCP Extensions』
RFC-1661	『The Point-to-Point Protocol (PPP)』
RFC-1994	『PPP Challenge Handshake Authentication Protocol (CHAP)』

標準/RFC - PPPoE	タイトル
RFC-2516	『A Method for Transmitting PPP Over Ethernet (PPPoE)』
RFC-4679	『DSL Forum Vendor-Specific RADIUS Attributes』

標準/RFC - L2TP	タイトル
RFC-2661	『Layer two tunneling protocol "L2TP"』

MIB

MIB	MIB のリンク
—	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/cisco/web/support/index.html

第 6 章

Quality of Service（QoS）の導入

Quality of Service（QoS）機能は、優先順位アプリケーションとの間のトラフィックがネットワーク リソースを使用するプリファレンスを取得できるようにします。QoS アクションは、ポリシーマップを使用して導入されるサービスポリシーによって定義されます。QoS プロセス中、パケットは QoS 情報でカプセル化されます。 カプセル化は、QoS アカウンティング機能によってモニタリングされ、説明されています。

パラメータ化された QoS（PQoS）は、トラフィックのプライオリティがトラフィックによって伝送されるデータ特性に基づく QoS の別の形式です。

BNG は、複数の動的なテンプレートを使用して適用される複数の QoS ポリシーマップのマージ、および単一の加入者での実装をサポートします。

この章では、QoS の導入について説明します。内容は次のとおりです。

Quality of Service（QoS）の概要

Quality of Service（QoS）は、時間的な制約があるネットワーク トラフィックと、VoIP サービス、ビデオのライブ ストリーミング、データベースへの優先アクセスなどのミッションクリティカルなアプリケーションに順位付けする方法です。 QoS が提供する機能により、このようなアプリケーションは少ない遅延で十分な帯域幅を受け取ることが可能になり、データ損失が減少します。

QoS 機能は、優先順位の高いパケットの優先転送を実行します。 これを行うために、ネットワーク全体のデータ転送パスのすべてのルータで、パケットは識別、分類、および順位付けされます。結果として、プライオリティ アプリケーションは必要なリソースを取得し、同時に、他のアプリケーションはネットワークにアクセスします。

QoS 機能は、既存のリソースの効率的な使用をイネーブルにすることによってサービス プロバイダーにコスト上の利点を提供し、反動的な拡張または過剰なネットワーク プロビジョニングを行わずに、必要なサービス レベルを保障します。 また、QoS は、さまざまなネットワーク アプリケーションから信頼できるサービスを取得するとカスタマー エクスペリエンスも向上します。

BNG がエッジ ルータに存在し、加入者はそこに直接接続するため、BNG で QoS を導入することは理想的です。 BNG の独自機能の 1 つが QoS アカウンティングです。 この機能によって、BNG は、QoS カプセル化情報を収集して RADIUS サーバに報告できます。 詳細については、QoS アカウンティング，（239 ページ）を参照してください。

QoS の導入には、次の 3 つのコンポーネントがあります。

- クラスマップ：一致ルールに基づいて、ビデオ、データ、VoIP などのさまざまなトラフィック形式を分類します。
- ポリシーマップ：分類されたトラフィックに適用される QoS アクションを定義します。 これは、以前にクラスマップに定義されたクラスを参照します。これらのポリシーマップは、QoS マップとも呼ばれます。 ポリシーマップで定義されたアクションは、トラフィックの優先順位付けおよび帯域割り当てを実行します。
- サービス ポリシー：BNG で、以前に定義されたポリシーマップを接続ポイントと方向に関連付けます。 接続ポイントは、QoS 接続ポイント，（260 ページ）に記載されています。ポリシーに可能な 2 方向は、入力および出力です。 ポリシーの方向は、接続ポイントに関連しています。

BNG は、QoS の導入について 2 つのレベルの階層型ポリシー（親ポリシーと子ポリシー）をサポートします。 サービス プロバイダーの設定に基づいて、QoS ポリシーは、次の方法で BNG で定義され、適用されます。

- CLI から QoS ポリシーを定義し、適用します。サービスポリシーの設定および動的なテンプレートを使用した加入者設定の適用，（225 ページ）を参照してください。
- CLI で QoS ポリシーを定義します。ただし、RAIDUS から適用します。サービスポリシーの設定および RADIUS を介した加入者設定の適用，（223 ページ）を参照してください。
- RAIDUS から QoS ポリシーを定義し、適用します。パラメータ化された QoS，（228 ページ）とも呼ばれます。

制約事項

加入者の入力または出力 QoS にポリシング、シェーピング、帯域幅、または WRED アクションが含まれる場合、アクティブ：スタンバイ バンドル インターフェイスの使用が推奨されます。ロードシェアリングは回避する必要があります。

サービスポリシーの設定および RADIUS を介した加入者設定の適用

CLI コマンドを使用して QoS ポリシーを導入するには、次の作業を実行します。 この作業では、RADIUS サーバから加入者設定を適用します。

手順の概要

1. **configure**
2. **policy-map type qos** *q_in*
3. **class class-default**
4. **service-policy** *q_child_in*
5. **policy-map type qos** *q_out*
6. **class class-default**
7. **service-policy** *q_child_out*
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type qos** *q_in* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_in`	QoS タイプのポリシーマップを設定します。
ステップ 3	**class class-default** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	親 class-default クラスを設定または変更します。 （注） 親ポリシーの class-default クラスは 1 つだけ設定できます。 他のトラフィック クラスは設定しないでください。
ステップ 4	**service-policy** *q_child_in* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy q_child_in`	最上位 class-default クラスに最下位ポリシーを適用します。

	コマンドまたはアクション	目的
ステップ 5	**policy-map type qos** *q_out* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_out`	QoS タイプのポリシーマップを設定します。
ステップ 6	**class class-default** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	親 class-default クラスを設定または変更します。 （注） 親ポリシーの class-default クラスは 1 つだけ設定できます。 他のトラフィック クラスは設定しないでください。
ステップ 7	**service-policy** *q_child_out* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy q_child_out`	最上位 class-default クラスに最下位ポリシーを適用します。

ステップ 8	次のいずれかのコマンドを使用します。
	
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CLI による加入者ポリシーの設定および RADIUS を介した適用：例

```
configure
policy-map type qos q_in
class class-default
end

\\the following procedure is ran in RADIUS
     Service-Type = Outbound-User
     Cisco-avpair = "ip:keepalive=protocol arp attempts 5 interval 15",
     Cisco-avpair = "ipv4:ipv4-mtu=750",
     Cisco-avpair = "ipv4:ipv4-unnumbered=Loopback0",
     Cisco-avpair = "subscriber:sub-qos-policy-in=q_in",
     Cisco-avpair = "subscriber:sub-qos-policy-out=q_out",
     Idle-Timeout = 1000,
     Session-Timeout = 5000
```

サービスポリシーの設定および動的なテンプレートを使用した加入者設定の適用

CLI コマンドを使用して QoS ポリシーを導入するには、次の作業を実行します。 この作業では、加入者設定は動的なテンプレートを使用して適用されます。

手順の概要

1. **configure**
2. **policy-map type qos** *q_in*
3. **class class-default**
4. **service-policy** *q_child_in*
5. **policy-map type qos** *q_out*
6. **class class-default**
7. **service-policy** *q_child_out*
8. **dynamic-template type ppp** *dynamic_config*
9. **service-policy input** *q_in*
10. **service-policy output** *q_out*
11. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type qos** *q_in* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_in`	入力方向でポリシーマップを設定します。
ステップ 3	**class class-default** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	親 class-default クラスを設定または変更します。 （注） 親ポリシーの class-default クラスは 1 つだけ設定できます。他のトラフィック クラスは設定しないでください。
ステップ 4	**service-policy** *q_child_in* 例： `RP/0/RSP0/CPU0:router(config)# service-policy q_child_in`	入力方向のサービス ポリシーを設定します。 （注） q_in および q_out ポリシーマップは、親ポリシー マップです。
ステップ 5	**policy-map type qos** *q_out* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_out`	出力方向のポリシーマップを設定します。 （注） q_in および q_out ポリシーマップは、親ポリシー マップです。
ステップ 6	**class class-default** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	親 class-default クラスを設定または変更します。 （注） 親ポリシーの class-default クラスは 1 つだけ設定できます。他のトラフィック クラスは設定しないでください。
ステップ 7	**service-policy** *q_child_out* 例： `RP/0/RSP0/CPU0:router(config)# service-policy q_child_out`	最上位 class-default クラスに最下位ポリシーを適用します。 （注） q_in および q_out ポリシーマップは、親ポリシー マップです。
ステップ 8	**dynamic-template type ppp** *dynamic_config* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp dynamic_config`	PPP タイプの動的なテンプレートを設定し、動的なテンプレートを使用して設定を適用します。

	コマンドまたはアクション	目的
ステップ 9	**service-policy input** *q_in* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input q_in`	入力方向でサービスポリシーを設定します。
ステップ 10	**service-policy output** *q_out* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input q_out`	出力方向でサービスポリシーを設定します。

ステップ 11	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CLI を介した加入者ポリシーの設定および動的なテンプレートを使用した加入者の適用：例

```
configure
policy-map type qos q_in  // policy-map input direction
class class-default
end

configure
policy-map type qos q_out  // policy-map output direction
class class-default
end

// applying configuration through dynamic-template
configure
dynamic-template type ppp dynamic_policy
service-policy input q_in
service-policy output q_out
end
```

パラメータ化された QoS

パラメータ化された Quality of Service（PQoS）は、必要なネットワーク帯域幅を予約することによってそのネットワーク アプリケーションの信頼性の高いパフォーマンスを保証します。この場合、優先順位付けは、パケットで伝送されるデータのタイプに基づきます。

標準 QoS では、パケットの重要性は、定義されているプライオリティ レベルに基づきます。ビデオ パケットと非同期データ転送パケットに同じプライオリティ レベルを定義できます。この場合、ルータは、両方のパケットを等しく重要視します。結果として、帯域幅の競合により、ビデオが低下する可能性があります。

一方、PQoS では、パケットの重要性は、パケットで伝送されるデータの特性やパラメータに基づきます。たとえば、PQoS でビデオ パケットの専用の帯域幅を提供できます。非同期データ トラフィックの重い負荷がネットワークに導入される場合でも、PQoS は、リアルタイム ストリーミングを必要としない他のデータ ストリームよりもビデオ パケットが優先されることを保証します。

パラメータ化された QoS は、QoS ポリシーマップ ベースのベンダー固有属性（VSA）を定義、変更、または削除できます。VSA は、RADIUS サーバからダウンロードされます。パラメータ化された QoS ポリシーの属性は、フィルタリングされ、ポリシー オブジェクト ライブラリに渡されます。後者では、属性を解析し、ポリシー オブジェクトに変換します。VSA は、加入者セッションに適用される 2 つのレベルの階層型ポリシーを定義します。QoS VSA の形式は、次のとおりです。

```
AVPair: qos-policy-in=add-class(sub,<parent-class, child-class>,<action-list>)
AVPair: qos-policy-out=add-class(sub,<parent-class, child-class>,<action-list>)
AVPair: qos-policy-in=remove-class(sub,<parent-class, child-class>)
AVPair: qos-policy-out=remove-class(sub,<parent-class, child-class>)
```

値は次のとおりです。

- 「sub」は、定数文字列で、加入者の現在のポリシーが変更されることを示します。
- <class-list> は、追加または削除されたクラスの階層（親クラス、子クラスなど）を示します
- <action-list> は、追加されるクラスで適用される QoS アクションを示します。

QoS パラメータとその構文の詳細については、RADIUS によるパラメータ化された QoS ポリシーの設定，（234 ページ）の「パラメータ化された *QoS* 構文」を参照してください。

加入者のパラメータ化された QoS ポリシーが RADIUS サーバから初めてダウンロードされるときに、VSA はポリシーを最初から構築するために使用されます。 ポリシーが加入者に適用された後、RADIUS サーバからその加入者にダウンロードされた新規または変更された VSA は、自動的にすでに適用されているポリシーを変更します。

RADIUS サーバからパラメータ化された QoS ポリシーを導入するには、RADIUS によるパラメータ化された QoS ポリシーの設定，（234 ページ）を参照してください。

許可変更（CoA）を使用すると、パラメータ化された QoS で以前に設定されたクラスマップを変更して、サービスポリシーを更新できます。 変更では、既存クラスの削除、または新規クラスの追加を実行できます。 サービスポリシーの更新を行うには、CoA によるサービスポリシーの変更，（237 ページ）を参照してください。

パラメータ化された QoS 構文

パラメータ化された QoS 構文

QoS アクション パラメータ	修飾子	コマンド
形状	QoS アクション	shape(<rate-in-kbps>)
	CLI 相当	shape average <shape-rate> <kbps>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default),shape(14700))
形状のパーセンテージ	QoS アクション	Shape-rpct(<rate-in-pct>)
	CLI 相当	shape average percent < rate-in-pct >
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default),shape-pct(25))
ポリシング（バリアント 1）	QoS アクション	police(<conform-rate-in-kbps>, <conform-burst-in-kBytes>, <exceed-rate-in-kbps>, <exceed-burst-in-kbytes>, <conform-action>, <exceed-action>, <violate-action>)

QoS アクション パラメータ	修飾子	コマンド
	CLI 相当	police rate <conform-rate> <kbps> burst <conform-burst> <kbps> peak-rate <exceed-rate> exceed-burst <exceed-burst> conform-action <action> exceed-action <action> violate-action <action>
	RADIUS 相当：例	qos-policy-in:add-class(sub,(class-default, voip),police(2000,2000, 4000, 4000,transmit, set-ipprec(< precedence>), drop))
ポリシング（バリアント 2）	QoS アクション	Police (<conform-rate-in-kbps>)
	CLI 相当	police rate <kbps>
	RADIUS 相当：例	qos-policy-in:add-class(sub,(class-default, voip), police(200000))
ポリシングのパーセンテージ（バリアント 1）	QoS アクション	police-rpct(<conform-rate-in-pct>, <conform-burst-in-us>, <exceed-rate-in-pct>, <exceed-burst-in-us>, <conform-action>, <exceed-action>, <violate-action>)
	CLI 相当	police rate percentage <pct> burst <conform-burst> < us> peak-rate percentage<pct> exceedburst <exceed-burst> conform-action <action> exceed-action <action> violate-action <action>
	RADIUS 相当：例	qos-policy-in:add-class(sub,(class-default, voip),police-rpct(20,20, 40, 40,transmit, set-ipprec(< precedence>), drop))

QoS アクション パラメータ	修飾子	コマンド
ポリシングのパーセンテージ（バリアント 2）	QoS アクション	Police-rpct(<conform-rate-in-pct>
	CLI 相当	police rate percentage <pct>
	RADIUS 相当：例	qos-policy-in:add-class(sub,(class-default, voip), police-rpct(20))
IP Precedence の設定	QoS アクション	set-ip-prec(<precedence>)
	CLI 相当	set precedence <precedence>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-ip-prec(5))
CoS の設定	QoS アクション	set-cos(<cos-val>)
	CLI 相当	set cos <cos-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-cos(5))
最小帯域幅	QoS アクション	bw-abs(<bw-in-kbps>)
	CLI 相当	bandwidth <bw-in-kbps>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,video),bw-abs(2000))
最小帯域幅のパーセンテージ	QoS アクション	bw-pct(<bw-in-pct>)
	CLI 相当	bandwidth percent <pct>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,video),bw-abs(2000))
帯域幅の残りのパーセンテージ	QoS アクション	bw-rpct(<pct>)

QoS アクション パラメータ	修飾子	コマンド
	CLI 相当	bandwidth remaining percent <pct>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip),bw-rpct(33))
IP DSCP の設定	QoS アクション	set-ip-dscp(<dscp-val>)
	CLI 相当	Set dscp <dscp-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-ip-dscp(46))
パケットのキュー制限	QoS アクション	queue-limit(<qlimit-in-packets>)
	CLI 相当	queue-limit <val> < packets>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip),queue-limit(64))
US のキュー制限	QoS アクション	queue-limit-us(<qlimit-in-us>)
	CLI 相当	queue-limit <val> <us>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip),queue-limit-us(240))
DSCPベースの WRED	QoS アクション	random-detect-dscp(<dscp>, <min-thresold>, <max-threshold>, <probability>)
	CLI 相当	random-detect dscp <dscp-val> < Min-thresh> <Kbytes> <max-thresh> <Kbytes> probability < probability-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), random-detect-dscp (24, 25000, 35000))
優先順位ベースの WRED	QoS アクション	random-detect-prec (<precedence>, <min-thresold>, <max-threshold>, <probability>)
	CLI 相当	random-detect precedence <prec-val> < Min-thresh> <Kbytes> <max-thresh> <Kbytes> probability < probability-val>

QoS アクション パラメータ	修飾子	コマンド
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), random-detect- (24, 25000, 35000))
QoS グループの設定	QoS アクション	set-qos-grp(<group-val>)
	CLI 相当	set qos-group <qos-group-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-qos-grp (24))
プライオリティ レベル	QoS アクション	pri-level(<priority-level>)
	CLI 相当	priority level <priority-level>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default, voip), pri_level(1))
廃棄クラスの設定	QoS アクション	set-dclass(<discard-class-val>)
	CLI 相当	set discard-class <discard-class-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-dclass (4))
MPLS EXP topmost ビットの設定	QoS アクション	set-mpls-exp-topmost (<mpls-exp- topmost-val>)
	CLI 相当	set mpls experimental topmost <mpls-exp- topmost-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-mpls-exp-topmost (4))
MPLS exp imposition ビットの設定	QoS アクション	set-mpls-exp- imposition (<mpls-exp-imposition-val>)
	CLI 相当	set mpls experimental imposition <mpls-exp- imposition-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-mpls-exp-imposition (4))

QoS アクション パラメータ	修飾子	コマンド
トンネルの優先順位の設定	QoS アクション	set-tunnel-prec(<prec-val>)
	CLI 相当	set precedence tunnel <precedence-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-tunnel-prec(4))
トンネルDSCPの設定	QoS アクション	set-tunnel-dscp (<dscp-val>)
	CLI 相当	set dscp tunnel <dscp-val>
	RADIUS 相当：例	qos-policy-out:add-class(sub,(class-default,voip), set-tunnel-dscp(4))

RADIUS によるパラメータ化された QoS ポリシーの設定

RADIUS サーバを使用して、パラメータ化された QoS ポリシーを導入し、加入者設定を適用するには、次の作業を実行します。 次の手順は、各加入者またはサービス プロファイルの RADIUS サーバで実行されます。

（注） パラメータ化されたQoS設定では、ポリシーマップはCLIで定義されません。 また、RADIUS によって送信された設定に基づいて動的に作成されます。 この手順は、RADIUS のユーザ設定の一部として、RADIUSサーバに適用されます。 ポリシーマップの結果は、ユーザ プロファイルがコントロール ポリシーの認証または許可アクションを実行した後にダウンロードされるときに、加入者に適用されます。 クラスマップは、CLIを使用して設定される必要があります。 このために、*classes voice_in*、*video_in*、*data_in*、*video_out*、*voice_out*、および *data_out* は、別々に設定されます。

手順の概要

1. **Cisco-AVPair** = *"ip:qos-policy-in=add-class(sub, (class-default),police(2000))"*
2. **Cisco-AVPair**+= *"ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"*
3. **Cisco-AVPair**+= *ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"*
4. **Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"*
5. **Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"*
6. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"*
7. **Cisco-AVPair** += *"ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"*
8. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"*
9. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"*
10. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,class-default))"*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**Cisco-AVPair** = *"ip:qos-policy-in=add-class(sub, (class-default),police(2000))"* 例： `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default),police(2000))"`	ポリシング アクション パラメータの入力方向で、cisco-avpair クラスマップを設定します。
ステップ 2	**Cisco-AVPair**+= *"ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"* 例： `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"`	ポリシング アクション パラメータの入力方向で、cisco-avpair クラスマップを設定します。
ステップ 3	**Cisco-AVPair**+= *ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"* 例： `Cisco-AVPair = ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"`	ポリシング アクション パラメータの入力方向で、cisco-avpair クラスマップを設定します。
ステップ 4	**Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"* 例： `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"`	ポリシング アクション パラメータの入力方向で、cisco-avpair クラスマップを設定します。

	コマンドまたはアクション	目的
ステップ 5	**Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"* 例： `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"`	QoS の設定アクション パラメータの入力方向で、cisco-avpair クラスマップを設定します。
ステップ 6	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"* 例： `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"`	形状アクション パラメータの出力方向で、cisco-avpair クラスマップを設定します。
ステップ 7	**Cisco-AVPair** += *"ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"* 例： `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"`	US のキュー制限アクション パラメータの出力方向で、cisco-avpair クラスマップを設定します。
ステップ 8	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"* 例： `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"`	US のキュー制限および形状アクション パラメータの出力方向で、cisco-avpair クラスマップを設定します。
ステップ 9	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"* 例： `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"`	帯域幅アクション パラメータの出力方向で、cisco-avpair クラスマップを設定します。
ステップ 10	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,class-default))"* 例： `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,class-default))"`	クラス アクション パラメータの出力方向で、cisco-avpair クラスマップを設定します。 （注） RADIUS によって設定および適用できる QoS アクション パラメータの完全なリストについては、パラメータ化された QoS 構文，（229 ページ）の「パラメータ化された *QoS* 構文」の項を参照してください。

RADIUS によって定義され適用された、パラメータ化された加入者ポリシーの設定：例

```
  Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default),police(2000))"
  Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1),
 police(256))"
  Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2),
 police(1000))"
  Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"

  Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,class-default),
set-qos-grp(7))"
  Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"
  Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,voice_out),
pri-level(1),queue-limit-us(10000))"
  Cisco-AVPair += "ip:qos-policy-out=add-class(sub,
(class-default,video_out),queue-limit-us(30000), shape(2000))"
  Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"

  Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,class-default))"
```

CoA によるサービス ポリシーの変更

CoA によってサービス ポリシーを変更するには、次の作業を実行します。

（注） CoA をサポートする Web ポータルまたは RADIUS サーバでは、このタスクの手順に対応する Cisco VSA で CoA 要求が生成されるように設定する必要があります。

手順の概要

1. **qos-policy-out** *remove-class(sub, (class-default, voip))*
2. **qos-policy-out** *add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))*
3. **qos-policy-out** *add-class(sub, (class-default, data), shape(400),set-ip-prec(1))*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**qos-policy-out** *remove-class(sub, (class-default, voip))* 例： `qos-policy-out=remove-class(sub, (class-default, voip))`	クラス マップを削除します。ここの voip は、加入者に対して以前設定したパラメータ化された QoS から削除されるクラスです。
ステップ 2	**qos-policy-out** *add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))* 例： `qos-policy-out=add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))`	クラス マップを追加します。ここの video は、加入者に対して以前設定したパラメータ化された QoS に追加されるクラスです。

	コマンドまたはアクション	目的
ステップ 3	**qos-policy-out** *add-class(sub, (class-default, data), shape(400),set-ip-prec(1))* 例： `qos-policy-out=add-class(sub, (class-default, data), shape(400),set-ip-prec(1))`	形状の qos-policy-out を設定し、IP precedence パラメータを設定します。

CoA によるサービス ポリシーの変更：例

```
//Policy-map configuration before CoA
policy-map __sub_5e311c4f_child1
 class voip
  priority level 1
  police rate 10000 kbps burst 8 kbytes
  !
!
class video
  priority level 1
  police rate 10000 kbps burst 16 kbytes
  !
!
class data
  shape average 80000 kbps
!
class class-default
!
end-policy-map
!
policy-map __sub_5e311c4f
 class class-default
  service-policy __sub_5e311c4f_child1
  shape average 100000 kbps
!
end-policy-map
!

//Modifying Service Policy through CoA
qos-policy-out=remove-class(sub, (class-default, voip))
qos-policy-out=add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))
qos-policy-out=add-class(sub, (class-default, data), shape(400),set-ip-prec(1))

//Policy-map configuration after CoA looks like:
 policy-map __sub_ffffffec1a37f_child1
class video
priority level 2
  bandwidth percent 50
  police rate 10000 kbps burst 16 kbytes
  !
!
class data
  shape average 400 kbps
  set precedence 1
!
class class-default
!
end-policy-map
!
policy-map __sub_ffffffec1a37f
class class-default
  service-policy __sub_ffffffec1a37f_child1
```

```
  shape average 100000 kbps
!
end-policy-map
!
```

QoS アカウンティング

QoS オーバーヘッド アカウンティング機能により、パケットに QoS を適用するときに、BNG はさまざまなカプセル化タイプを考慮することができます。 ATM オーバーヘッド アカウンティングにより、BNG は加入者線上の ATM カプセル化を考慮することができます。 セル分割で追加されたオーバーヘッドも考慮します。このアカウンティングを使用すれば、サービスプロバイダーは加入者線でのオーバーランを防止することができ、BNGは加入者トラフィックに割り当てられる実際の帯域幅に対して QoS 機能を実行できるようになります。 ATM オーバーヘッドのカプセル化の詳細を次の表に示します。

表 *3*：*ATM* オーバーヘッドのカプセル化の詳細

CPE カプセル化に対する DSLAM		**ALE タグ（RFC 4679）**		
CLI オプション	**オーバーヘッド（バイト）**	**データリンク**	**カプセル化 1**	**カプセル化 2**
snap-pppoa	12	AAL5	該当なし	PPPoA LLC（1）
mux-pppoa	10	AAL5	該当なし	PPPoA ヌル（2）
snap-1483routed	18	AAL5	タグなしイーサネット	IPoA LLC（3）
mux-1483routed	8	AAL5	タグなしイーサネット	IPoA ヌル（4）
snap-rbe	28	AAL5	タグなしイーサネット	FCS（6）のない Ethernet over AAL5 LLC
snap-dot1q-rbe	32	AAL5	シングルタグイーサネット	FCS（6）のない Ethernet over AAL5 LLC
mux-rbe	24	AAL5	タグなしイーサネット	FCS（8）のない Ethernet over AAL5 ヌル
mux-dot1q-rbe	28	AAL5	シングルタグイーサネット	FCS（8）のない Ethernet over AAL5 ヌル

QoS オーバーヘッド アカウンティングをイネーブルにするには、QoS アカウンティングの設定,（240 ページ）を参照してください。

QoS アカウンティングの設定

QoS レイヤ 2 オーバーヘッド アカウンティングをイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type** [**ppp**|**ip-subscriber**|**service**]*name*
4. **qos-account** [**AAL5**| **user-defined**] [**mux-1483routed** | **mux-dot1q-rbe** | **mux-pppoa** | **mux-rbe** | **snap-1483routed** | **snap-dot1q-rbe** | **snap-ppoa** | **snap-rbe**]
5. **exit**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 3	**type** [**ppp**|**ip-subscriber**|**service**]*name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp p1`	適用する必要がある動的なテンプレートのタイプを指定します。 3 つのタイプは次のとおりです。 • PPP • IP 加入者 • サービス

	コマンドまたはアクション	目的
ステップ 4	**qos-account** [**AAL5**| **user-defined**] [**mux-1483routed** | **mux-dot1q-rbe** | **mux-pppoa** | **mux-rbe** | **snap-1483routed** | **snap-dot1q-rbe** | **snap-ppoa** | **snap-rbe**] 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# qos-account AAL5 snap-rbe`	L2 QoS オーバーヘッドアカウンティングを定義します。 mux-1483routed、snap-rbe などのさまざまなキーワードは、DSLAM と CPE 間で使用可能な異なるカプセル化を定義します。 キーワードの詳細については、表 3：ATM オーバーヘッドのカプセル化の詳細，（239 ページ）を参照してください。
ステップ 5	**exit** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# exit`	現在のモードを終了します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

QoS アカウンティングの設定：例

```
configure
dynamic-template type ppp p1
qos account AAL5  mux-1483routed
service-policy input input_1
end
```

共有ポリシー インスタンスのサポート

共有ポリシー インスタンス（SPI）は、BNG サブインターフェイスおよびバンドル サブインターフェイスのグループ間で単一の QoS リソース セットの割り当てを可能にし、サブインターフェイス グループ、複数のイーサネット フロー ポイント（EFP）、またはバンドル インターフェイスで QoS リソースを共有します。

SPI を使用して、QoS ポリシーの 1 つのインスタンスを複数のサブインターフェイスで共有し、サブインターフェイスのシェーピングを 1 つのレートに集約できます。 QoS ポリシーのインスタンスを共有するすべてのサブインターフェイスは、同じ物理インターフェイスに属している必要があります。 QoS ポリシーのインスタンスを共有するサブインターフェイス数の範囲は、2 からポートのサブインターフェイスの最大数までです。

バンドル インターフェイスの場合、ハードウェア リソースはバンドル メンバごとに複製されます。 共通の共有ポリシー インスタンスを使用し、Link Aggregation Control Protocol（LAG）バンドルで設定されたサブインターフェイスは、すべて同じメンバリンクにロードバランシングされる必要があります。

バンドル EFP にポリシーが設定されている場合、バンドルのメンバ リンクごとにポリシーのインスタンスが 1 つ設定されます。 同じバンドルの複数のバンドル EFP 間で SPI を使用する場合、バンドルのメンバ リンクごとにポリシーの共有インスタンスが 1 つ設定されます。 デフォルトでは、バンドルのロード バランシング アルゴリズムによってハッシュが使用され、（バンドル EFP から送信される必要のある）トラフィックをバンドル メンバ間に分散させます。 1 つまたは複数の EFP のトラフィックを、複数のバンドル メンバ間に分散させることができます。 複数の EFP に、SPI を使用して一緒にシェーピングまたはポリシングされる必要があるトラフィックがある場合、同じ共有ポリシーのインスタンスに属するすべての EFP へのトラフィックに対して、バンドル ロード バランシングで同じバンドル メンバを選択する（ハッシュ選択）ように設定する必要があります。 これによって、同じポリシーの共有インスタンスを持つすべての EFP に向かうトラフィックで、同じポリサーまたはシェーパー インスタンスが使用されます。

BNGは、親および子ポリシーを含む完全な階層型ポリシーマップを設定します。オプションで、SPI の名前を定義し、次のように、適切な動的テンプレートに接続するか、RADIUS からダウンロードできます。

- CLI を介して設定され、動的なテンプレートによって適用されるポリシー
- CLI を介して設定され、RADIUS によって適用されるポリシー

制約事項

次の制約事項が、共有ポリシー インスタンスの使用に適用されます。

- SPI は、非バンドル インターフェイスの加入者にはサポートされません。
- SPI は、パラメータ化された QoS（PQoS）ではサポートされません。 PQoS の設定で、SPI 名がある場合、それは無視されます。
- CoA によって変更された SPI は、加入者でサポートされません。

動的なテンプレートを使用した入力または出力方向での SPI を持つポリシーの設定

動的なテンプレートを使用して入力および出力方向で共有ポリシー インスタンスを持つポリシーを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **policy-map** *policy_map_name*
3. **class** {*class_name* | **class-default** | } [**type qos**]
4. **service-policy** *service_policy_name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
6. **policy-map** *policy_map_name*
7. **class** {*class_name* | **class-default** | } [**type qos**]
8. **police rate** *value*
9. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
10. **dynamic-template type ipsubscriber** *dynamic_template_name*
11. **service-policy** {**input** |**output**}*policy_map_name* [**shared-policy-instance** *instance_name*]
12. **service-policy** {**input** |**output**}*policy_map_name* [**shared-policy-instance** *instance_name*]
13. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map** *policy_map_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map policy1`	サービス ポリシーを指定するために 1 つまたは複数のインターフェイスに対応付けることができるポリシーマップを作成または修正し、ポリシーマップ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**class** {*class_name* | **class-default** | } [**type qos**] 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	作成または変更するポリシーのクラス名を指定し、ポリシー マップ クラス コンフィギュレーション モードを開始します。 この例では、トラフィック ポリシー policy1 のデフォルト クラスに対するトラフィック ポリシーの設定を示しています。 デフォルト クラスの名前は、class-default です。
ステップ 4	**service-policy** *service_policy_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy policy1_child`	ポリシー マップを入力または出力インターフェイスに適用します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 6	**policy-map** *policy_map_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map policy1_child`	サービス ポリシーを指定するために 1 つまたは複数のインターフェイスに対応付けることができるポリシー マップを作成または修正し、ポリシーマップ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 7	**class** {*class_name* | **class-default** | } [**type qos**] 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	作成または変更するポリシーのクラス名を指定し、ポリシー マップ クラス コンフィギュレーション モードを開始します。 この例では、トラフィック ポリシー policy1 のデフォルト クラスに対するトラフィック ポリシーの設定を示しています。 デフォルト クラスの名前は、class-default です。
ステップ 8	**police rate** *value* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 1024`	トラフィック ポリシングを設定し、ポリシー マップ ポリシング コンフィギュレーション モードを開始します。値は、認定情報レートを表し、1 ～ 4294967295 の範囲です。

ステップ 9	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 10	**dynamic-template type ipsubscriber** *dynamic_template_name* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp PTA_TEMPLATE_1`	ipsubscriber タイプの動的なテンプレートを作成します。

	コマンドまたはアクション	目的
ステップ 11	**service-policy** {**input** |**output**}*policy_map_name* [**shared-policy-instance** *instance_name*] 例： `RP/0/RSP0/CPU0:router(config)# service-policy input policy1 shared-policy-instance spi_1`	インターフェイスのサービス ポリシーとして使用する入力インターフェイスまたは出力インターフェイスにポリシー マップを付加します。 この例では、トラフィック ポリシーでそのインターフェイスに送信されるすべてのトラフィックを評価します。
ステップ 12	**service-policy** {**input** |**output**}*policy_map_name* [**shared-policy-instance** *instance_name*] 例： `RP/0/RSP0/CPU0:router(config)# service-policy output policy1 shared-policy-instance spi_2`	インターフェイスのサービス ポリシーとして使用する入力インターフェイスまたは出力インターフェイスにポリシー マップを付加します。 この例では、トラフィック ポリシーでそのインターフェイスから送信されるすべてのトラフィックを評価します。

ステップ 13	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

動的なテンプレートを使用した入力または出力方向での SPI を持つポリシーの設定：例

```
configure
policy-map policy1
class class-default
```

```
service-policy policy1_child
!!

policy-map policy1_child
class class-default
police rate 1024 kbps
!!

dynamic-template
type ppp PTA_TEMPLATE_1
service-policy input policy1 shared-policy-instance spi_1
service-policy output policy1 shared-policy-instance spi_2
commit
```

RADIUS を使用した入力または出力方向での SPI を持つポリシーの設定

RADIUS を使用して入力および出力方向で共有ポリシー インスタンスを持つポリシーを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **policy-map** *policy_map_name*
3. **class** {*class_name* | **class-default**} [**type qos**]
4. **service-policy** *service_policy_name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
6. **policy-map** *policy_map_name*
7. **class** {*class_name* | **class-default**} [**type qos**]
8. **police rate** *value*
9. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**policy-map** *policy_map_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map policy1`	サービス ポリシーを指定するために1つまたは複数のインターフェイスに対応付けることができるポリシーマップを作成または修正し、ポリシーマップ コンフィギュレーション モードを開始します。
ステップ 3	**class** {*class_name* | **class-default**} [**type qos**] 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	作成または変更するポリシーのクラス名を指定し、ポリシー マップ クラス コンフィギュレーション モードを開始します。 この例では、トラフィック ポリシー policy1 のデフォルト クラスに対するトラフィック ポリシーの設定を示しています。 デフォルト クラスの名前は、class-default です。
ステップ 4	**service-policy** *service_policy_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy policy1_child`	ポリシーマップを入力または出力インターフェイスに適用します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 6	**policy-map** *policy_map_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map policy1_child`	サービス ポリシーを指定するために1つまたは複数のインターフェイスに対応付けることができるポリシーマップを作成または修正し、ポリシーマップ コンフィギュレーション モードを開始します。
ステップ 7	**class** {*class_name* | **class-default**} [**type qos**] 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	作成または変更するポリシーのクラス名を指定し、ポリシー マップ クラス コンフィギュレーション モードを開始します。 この例では、トラフィック ポリシー policy1 のデフォルト クラスに対するトラフィック ポリシーの設定を示しています。 デフォルト クラスの名前は、class-default です。
ステップ 8	**police rate** *value* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 1024`	トラフィック ポリシングを設定し、ポリシー マップ ポリシング コンフィギュレーション モードを開始します。 値は、認定情報レートを表し、1 ～ 4294967295 の範囲です。

ステップ 9	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS を使用した入力または出力方向での SPI を持つポリシーの設定：例

```
configure
policy-map policy1
class class-default
service-policy policy1_child
!!

policy-map policy1_child
class class-default
police rate 1024 kbps
commit
!!

//In the USER file in RADIUS
RoadRunner_P1@Chasing1 Cleartext-Password := "LooneyTunes_P1"
cisco-avpair += "sub-qos-policy-in=policy1 shared-policy-instance spi_1",
cisco-avpair += "sub-qos-policy-out=policy1 shared-policy-instance spi_2",
Framed-Protocol += PPP,
Service-Type += Framed-User,
Fall-Through = no
```

次の作業

RADIUS のユーザ ファイルで次の手順を実行します。

```
RoadRunner_P1@Chasing1 Cleartext-Password := "LooneyTunes_P1"
    cisco-avpair += "sub-qos-policy-in=policy1 shared-policy-instance spi_1",
    cisco-avpair += "sub-qos-policy-out=policy1 shared-policy-instance spi_2",
    Framed-Protocol += PPP,
    Service-Type += Framed-User,
    Fall-Through = no
```

QoS ポリシーマップのマージ

複数の動的なテンプレートによって適用される複数の QoS ポリシーは、1 つの加入者にマージおよび実装できます。 ポリシーがマージされる順序は重要で、これによって動的なテンプレートで設定されるシーケンス番号の値が決定します。 ポリシーは、ポリシーマップを使用して導入されます。 新しい任意の **merge** キーワードは、動的なテンプレートのサブモードで **service-policy** コマンドとともに提供され、複数の動的なテンプレートで適用されるポリシーマップのマージを可能にします。

2 つ以上のポリシーマップをマージする場合、まず 2 つのポリシーマップを 1 つにマージして、マージされたポリシーマップを作成します。 次に、3 番目のポリシーマップを最初にマージしたポリシーマップとマージします。 マージされるすべてのポリシーマップが 1 つにマージされるまで続けます。 たとえば、ポリシーマップ p1、p2、p3、p4 がこの順序でマージされると仮定します。p1 と p2 が最初にマージされます（次に示すルールを使用）。 次に、p3 は、マージされたポリシーマップ<p1-p2>とマージされます。 最後に、p4 は、マージされたポリシーマップ<p1-p2-p3>とマージされ、マージされた最終的なポリシーマップになります。

2 つのポリシーマップをマージするためのルールは、次のとおりです。

- マージされるポリシーマップは、最初のポリシーマップ クラスの後ろに 2 番目のポリシーマップ クラスを追加することによって作成できます（デフォルト クラスを除く）。

- 両方のポリシーの下に同じクラスが設定されている場合（デフォルト クラスを除く）、2 番目のポリシーのクラスのインスタンス（その下に設定されたすべてのアクションを含む）は無視されます。
- 最初のポリシーの下のデフォルト クラスに子ポリシー アクション以外のアクションが含まれる場合、そのデフォルト クラスは、マージされたポリシーの最後に追加されます。 子ポリシー アクションが含まれる場合、2 番目のポリシーのデフォルト クラスは、マージされたポリシーの最後に追加されます。
- 子ポリシーが、両方のポリシーのデフォルト クラスの下に設定されている場合、2 つの子ポリシーは、上記のルールでマージされます。 マージされた子ポリシーは、マージされた親ポリシーのデフォルト クラスの下の子ポリシーとして適用されます。
- 子ポリシーが最初または 2 番目のポリシーのいずれか（両方ではない）のデフォルト クラスの下で設定される場合、マージされたポリシーのデフォルト クラスの下の子ポリシーとして（そのまま）適用されます。 デフォルト クラス以外のクラスの下の子ポリシーは、1 つにマージされません。

（注） マージされる 2 つのポリシーのシーケンス番号が同じに設定される場合、相互についてマージされる順序は任意で、プロセスの再起動後に変わることがあります。 このような設定は回避する必要があります。

ポリシーマップのマージのイネーブル化

複数の動的なテンプレートを使用して適用した複数の QoS ポリシーマップのマージをイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type service** *dynamic-template-name*
4. **service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**] [**merge** *seq_num*]
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ3	**type service** *dynamic-template-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service s1`	サービスのユーザ定義名で動的なテンプレートを作成します。

ステップ4	**service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**] [**merge** *seq_num*]
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input QoS1 merge 10`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy output QoS2 merge 20`
		サービスポリシーを動的なテンプレートに関連付け、複数のQoSポリシーのマージをイネーブルにします。

ステップ5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッショ

	コマンドまたはアクション	目的
		ンは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ポリシーマップのマージのイネーブル化：例

```
dynamic-template type service default-service
   service-policy input default-policy-in merge 100
   service-policy output default-policy-out merge 100
!
dynamic-template type service voip-service
   service-policy input voip-policy-in merge 20
   service-policy output voip-policy-out merge 30
!
dynamic-template type service vod-service
   service-policy input vod-policy-in merge 30
   service-policy output vod-policy-out merge 50
!
dynamic-template type service turbo-button-service
   service-policy input turbo-button-policy-in merge 10
   service-policy output turbo-button-policy-out merge 40
!
end

\\the following configuration explains the merging behavior of egress qos policies
policy-map type qos default-policy-out
  class class-default
      shape average 2 mbps
      bandwidth 512 kbps
    service-policy default-policy-child-out
  !
  end-policy-map

policy-map type qos default-policy-child-out
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
      set cos 5
    !
    class class-default
      shape average percent 20
      set cos 7
    !
    end-policy-map

policy-map type qos voip-policy-out
   class class-default
     service-policy voip-policy-child-out
   !
```

```
  end-policy-map

 policy-map type qos voip-policy-child-out
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
   class class-default
   !
   end-policy-map

policy-map type qos vod-policy-out
  class class-default
    service-policy vod-policy-child-out
  !
  end-policy-map


policy-map type qos vod-policy-child-out
  class vod-control
     priority level 1
     set cos 1
  !
  class vod-data
   priority level 2
   queue-limit 100 ms
  !
  class class-default
  !
   end-policy-map

policy-map type qos turbo-button-policy-out
  class class-default
     shape average 10 mbps
     bandwidth 2 mpbs
   !
   end-policy-map


\\after the default and voip services are enabled on a subscriber session

policy-map type qos  <merged-policy-1>   !! Name is generated internally. This is just an
example
  class class-default
      shape average 2 mbps
      bandwidth 512 kbps
      service-policy <merged-child-policy-1>
    !
   end-policy-map

policy-map type qos <merged-child-policy-1>
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
```

```
      set cos 5
     !
     class class-default
       shape average percent 20
       set cos 7
     !
     end-policy-map

\\after the turbo-button service is enabled

policy-map type qos <merged-policy-2>
   class class-default
     shape average 10 mbps
     bandwidth 2 mpbs
     service-policy <merged-child-policy-1> !! <merged-child-policy-1> is the same as before
 since the
                                                                          !! the
turbo-button-policy-out does not have any child policy
                                                                          !! to be merged.
   !

\\after the vod service is enabled

policy-map type qos  <merged-policy-3>
  class class-default
      shape average 10 mbps
      bandwidth 2 mbps
      service-policy <merged-child-policy-2>
    !
   end-policy-map

policy-map type qos <merged-child-policy-1>
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
  class vod-control
     priority level 1
     set cos 1
  !
  class vod-data
   priority level 2
   queue-limit 100 ms
  !
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
      set cos 5
    !
    class class-default
      shape average percent 20
      set cos 7
    !
    end-policy-map
```

BNG でサポートされる QoS 機能

BNG は次の QoS 機能をサポートします。

ポリシングおよびキューイングのサポート

BNG は、入出力トラフィック ポリサーを提供します。 BNG は、加入者セッションごとに既存のトラフィック ポリシング メカニズムもサポートします。 マーキング アクションを含む 1R2C および 2R3C ポリサーは、加入者ポリシーの親レベルでサポートされます。 絶対ポリシング レートのみが、加入者ポリシーの親レベルでサポートされます。 マーキング アクションを含む 1R2C および2R3C ポリサーは、加入者ポリシーの子レベルでサポートされます。 絶対およびパーセンテージ ベースの両方のポリシング レートは、加入者ポリシーの子レベルでサポートされます。

BNG は、物理ポート レベル、加入者セッション レベル、クラス レベル、および VLAN レベル（出力方向のみ）で、トラフィック シェーピングをサポートします。 システムは、加入者セッションに対する既存のキューイングアクションをすべてサポートします。加入者ポリシーの親レベルの最小帯域幅設定は、ブロックされます。加入者ポリシーにキューイングアクションがない場合、これらの加入者のトラフィックは S-VLAN のシェーピングに従ったままで、キューイングアクションがある場合、トラフィックは S-VLAN ポリシーのキューを通って出て行きます。そうでない場合、トラフィックはインターフェイスのデフォルトキューを通過します。 シェーピング アクションまたは帯域幅残量キューイング アクションは、フラット S-VLAN ポリシーに必須です。 S-VLAN フラット ポリシーと加入者ポリシーの親レベルでは、絶対シェープ レートのみがサポートされます。 ただし、シェーピング アクションと帯域幅残量キューイング アクションのみが加入者ポリシーの親レベルでサポートされ、すべてのキューイング アクションは加入者ポリシーの子レベルでサポートされます。

これらの追加のキューイング機能は、加入者に適用される出力ポリシーでサポートされます。

- ポリシーには、1 つの P1、1 つの P2、1 つの P3、および 5 つの標準プライオリティ キューを設定できます。
- ポリシーには、1 つの P1、2 つの P2、および 5 つの標準プライオリティ キューを設定できます。 P1 および P3 キューは複数のクラスで共有できますが、P2 キューは共有されません。

デフォルトのマーキング

BNG は、加入者セッションと併用する L3 インターフェイスでサポートされる既存の分類とマーキング オプションのすべてをサポートします。 BNG は、L2 マーキングへの L3 マーキングのマッピングもサポートします。 BNG は、ダウンストリーム PPPoE フレームの LAC での ToS から CoS へのマッピングもサポートし、802.1p と IP ToS フィールドをマークするメカニズムも提供します。 システムは、入力の加入者 QoS ポリシーに基づいて、L2TP パケットの柔軟な IP TOS マーキングを可能にします。 マーキングは、加入者ポリシーの親レベルと加入者ポリシーの子レベルでサポートされます。

QoS ポリシーの変更

BNG は、インサービス QOS ポリシー変更をサポートします。 加入者ポリシー（RADIUS 経由）、S-VLAN ポリシー（CLI 経由）およびポート サブレート ポリシー（CLI 経由）の変更もサポートされています。

L2 のカプセル化

PPPoE加入者の場合、QoS レート計算で使用されるL2のカプセル化サイズは、PPPoEタグでシグナリングされるラスト マイル カプセル化（加入者の自宅に対する DSLAM）に基づいて調整できる必要があります。

分類

BNG は、加入者セッションと併用する L3 インターフェイスでサポートされる既存の分類とマーキング オプションのすべてをサポートします。 BNGは、シングル タグおよびダブル タグのCOSの802.1p値に基づく入力の分類、いずれかの方向のDSCPに基づく分類、いずれかの方向のL3/L4 ACL に基づく分類、および外部 DSCP マーキングに基づく L2TPv2 トラフィックの分類もサポートします。

入力コア側インターフェイスの着信 L2TP パケットの分類は、パケットが MPLS タグ スタックとともに到達する場合でも、常に外部 IP フィールドに基づきます。

ポリシーの継承

次の表が該当するのは出力方向のみです。入力方向のサブレート ポリシーとS-VLANポリシーはサポートされません。

ポート	S-VLAN	サブスクライバ
サブレート ポリシー	ポリシーは設定されません。継承は、ポートのサブレート ポリシーによって形成されるトラフィックに制限されます。これは、ポリシーがS-VLANで設定されているかどうかに関係なく行われます。	加入者ポリシーが存在する場合は、最初に実行され、トラフィックはポートシェーパーに従います。
サブレート ポリシー	ポリシーが設定されます。 継承は、ポートのサブレート ポリシーによって形成されるトラフィックに制限されます。 これは、ポリシーがS-VLANで設定されているかどうかに関係なく行われます。	加入者ポリシーが存在する場合は、最初に実行され、その後、S-VLANポリシーが実行されます。 最後に、トラフィックはポートシェーパーに従います。
クラスデフォルトより多くのHQoS またはポリシー	ポリシー設定はブロックされ、ポート ポリシーが継承されます。	ポリシー設定はブロックされ、ポート ポリシーが S-VLAN を介して継承されます。
ポリシーの設定なし	ポリシーが設定されます。	加入者ポリシーが存在する場合は、最初に実行され、その後、S-VLANポリシーが実行されます。

QoS のない加入者

QoS が加入者で設定されていない場合、親 S-VLAN、またはポートで、加入者トラフィックは親の物理ポートのデフォルトキューを使用して出て行きます。

- 加入者は S-VLAN ポリシーに従い、S-VLAN ポリシーのキューが存在する場合はそれを使用して出て行きます。 S-VLAN ポリシーに独自のキューがない場合、加入者のトラフィックを含むすべての S-VLAN トラフィックは、物理インターフェイスのデフォルトキューを介して出て行きます。
- 加入者はポート ポリシーに従いますが、S-VLAN ポリシーには従いません。 S-VLAN の場合と同様に、加入者トラフィックはそれに従い、キューを使用します。
- 非ポートシェーパー ポリシーがポートに適用される場合、S-VLAN と加入者でのポリシーの適用は、ブロックされます。 このような場合、加入者トラフィックは、ポートに適用されたポリシーに従います。

制御パケットの処理

BNG は、PPP リンク制御プロトコル（LCP）パケットの処理で優先処理を行います。 制御パケットは、ユーザ設定せずに高いプライオリティで処理され、これらのパケットは、インターフェイスの入力出力の両方に適用される QoS ポリシーに従いません。 LAC のアップストリーム方向では、信頼できる CoS 値が必要な場合、信頼できる CoS に基づいてコア側のヘッダーを課すために PPP コマンドが提供されます。 したがって、ネットワークの制御パケットの優先処理が保障されます。

S-VLAN のシェーピングおよび統計情報

出力方向で、BNG は、加入者インターフェイス レベル、スタックされた仮想ローカルエリア ネットワーク（S-VLAN）、およびポート レベルの 3 つの異なるレベルでポリシーを設定する機能をサポートします。 出力 S-VLAN およびポートレベル ポリシーは、インターフェイス レベルで CLI を介して直接適用されます。 S-VLAN に QoS ポリシーを適用するには、S-VLAN でのポリシーの設定，（262 ページ）を参照してください。

加入者ポリシーは、動的なテンプレートまたは RADIUS を介してのみ適用できます。 出力の加入者ポリシーには、2 つのレベルのポリシーを指定できます。 S-VLAN およびポートレベル ポリシーには、フラット ポリシーのみ、クラス デフォルトのみ、シェーピング レートであるアクションのみを指定できます。 基本的に、シェーピングによって S-VLAN またはポートを最大レートに抑制する手段を提供します。

入力方向では、トラフィックは、加入者ポリシーが RADIUS または動的なテンプレートによって適用される加入者の入力ポリシーにのみ従います。

S-VLAN によるトラフィックには、加入者ポリシーによってすでに形成されている可能性がある多数の加入者へのトラフィックが含まれます。 S-VLAN シェーパーで統計情報を提供することは、最大容量に到達しているかどうかをモニタリングするために重要です。 加入者 QoS ポリシーとは異なり、ハードウェアには、この S-VLAN のシェーパーを通じて使用または送信されるパケット/バイトを直接追跡する機能はありません。 その他の統計情報とは異なり、BNG は、基盤となる加入者ポリシーの統計情報を集約することによって、S-VLAN QoS ポリシー関連の統計情報を提供

します。 統計情報は、他のすべてのインターフェイス タイプに対応した show コマンド（および適切な MIB）で表示されます。

S-VLAN は、次の条件をサポートします。

- QoS レートの変更。
- ポリシーのレベル数を変更する S-VLAN ポリシーの変更は拒否されます。
- 子レベルのクラスを追加または削除する 2 レベルの S-VLAN ポリシーの変更は拒否されます。
- 2 レベル ポリシーでの、子レベルのクラスの分類基準の変更は拒否されます。
- 2 レベル ポリシーおよびフラット ポリシーの両方での、アクションの追加または削除は拒否されます。

QoS 接続ポイント

次の表に、QoS 接続ポイントと定義およびアプリケーションのモードを示します。

QoS 接続ポイント	定義	アプリケーション	ポリシーのタイプ
ポート（サブレート ポリシー）	CLI/XML	CLI/XML	フラット：クラスデフォルトのみ
S-VLAN	CLI/XML	CLI/XML	フラット：クラスデフォルトのみ。 2 レベル：親のクラスデフォルトのみ、および子のすべての分類。
サブスクライバ	CLI/XML	動的なテンプレート	2 レベル：親のクラスデフォルトのみ、および子のすべての分類。
サブスクライバ	CLI/XML	RADIUS	2 レベル：親のクラスデフォルトのみ、および子のすべての分類。
サブスクライバ	RADIUS（パラメータ化された QoS）	RADIUS	2 レベル：親のクラスデフォルトのみ、および子のすべての分類。

サポートされていない設定は、ブロックされません。 S-VLAN ポリシーおよび加入者ポリシーでは、次の表に示されている以外の設定はブロックされます。

表 *4*：入力方向でサポートされる設定

	分類	アクション	レート
加入者の親レベル ポリシー	クラスデフォルトのみ	ポリシング、マーキング	絶対のみ
加入者の子レベル ポリシー	すべて、ベースライン制限あり	ポリシング、マーキング	絶対およびパーセンテージ

表 *5*：出力方向でサポートされる設定

	分類	アクション	レート
S-VLAN フラット ポリシー	クラスデフォルトのみ	すべて、必須の形成アクションあり	絶対のみ
S-VLAN の親レベル ポリシー	クラスデフォルトのみ	すべて、必須の形成アクションあり	絶対のみ
S-VLAN の子レベル ポリシー	すべて、ベースライン制限あり	すべて	絶対およびパーセンテージ
加入者の親レベル ポリシー	クラスデフォルトのみ	形成、帯域幅残量、ポリシング、マーキング	絶対のみ
加入者の子レベル ポリシー	すべて、ベースライン制限あり	すべて	絶対およびパーセンテージ

アクセス インターフェイスの VLAN ポリシー

BNGは、アクセスインターフェイスの入力および出力VLANポリシーをサポートします。 S-VLAN（加入者の親）ポリシーとは異なり、アクセスインターフェイスの VLAN ポリシーは、セッション ポリシーから継承されず、VLANポリシーに基礎となるセッション ポリシーからの参照帯域幅および統計情報の集約はありません。

S-VLAN ポリシーは、出力方向でのみサポートされます。 VLAN ポリシーは、入力方向と出力方向の両方でサポートされます。 設定された入力 VLAN ポリシーは、通常の VLAN ポリシーと同様に動作し、セッション ポリシーには影響しません。 **subscriber-parent** キーワードは、出力の S-VLAN ポリシーを設定するためにのみ使用されます。 出力 VLAN ポリシーが **subscriber-parent** キーワードで設定されている場合、ポリシーはS-VLANポリシーになり、セッション ポリシーで継承されます。 一方で、出力VLANポリシーが **subscriber-parent** キーワードなしで設定されている場合は、入力側と同様に動作し、セッションポリシーには影響しません。 詳細については、アクセス インターフェイスでの VLAN ポリシーの設定, （264 ページ）を参照してください。

この例は、アクセスインターフェイスに接続されたサンプル出力 VLAN ポリシーを示します。

```
service-policy output metering
```

この例は、アクセスインターフェイスに接続されたサンプル出力 S-VLAN ポリシーを示します。

```
service-policy output metering subscriber-parent
```

セッション レベルの入力 QoS ポリシーとアクセスインターフェイス レベルの入力 QoS ポリシー（**subscriber-parent** キーワードなし）は共存できます。 これらの 2 つのポリシー間に依存関係はなく、これらのポリシーの操作は全体的に互いに依存しません。 加入者セッションは同じアクセスインターフェイスで起動されますが、加入者セッション上の入力トラフィックは、アクセスインターフェイスに接続された入力 QoS ポリシーに従いません。 同様に、入力方向以外の加入者トラフィックは、アクセスインターフェイスの入力 QoS ポリシーに従います。

アクセスインターフェイス レベルの入力 QoS ポリシーには、加入者以外のトラフィックに対する QoS の統計情報があります。 加入者レベルの入力 QoS ポリシーには、加入者トラフィックのみの統計情報があります。 つまり、加入者トラフィックのみがアクティブな場合、アクセスインターフェイスに接続された入力 QoS ポリシーに統計情報は表示されません。

次の表は、入出力方向の VLAN および S-VLAN ポリシーのサポートの要約を示しています。

ポリシーの方向	**V-LAN** ポリシー（**subscriber-parent** キーワードなし）	**S-VLAN** ポリシー（**subscriber-parent** キーワードを使用）
入力	サポートあり	サポート対象外
出力	サポートあり	サポートあり

制約事項

次の制約事項が、**subscriber-parent** キーワードなしで使用される場合に、アクセスインターフェイスの VLAN ポリシーに適用されます。

- VLAN ポリシーは、ポリシーを使用してセッション起動する前に、アクセスインターフェイスに接続される必要があります。
- S-VLAN ポリシーのインプレース変更用に指定された制約事項は、VLAN ポリシーにも適用されます。 たとえば、VLANポリシーのインプレース変更は、レート変更のみをサポートします。

S-VLAN でのポリシーの設定

S-VLAN に QoS ポリシーを適用するには、次の作業を実行します。

（注）
- S-VLANポリシーは、ポリシーが加入者にインストールされる前にプロビジョニングされる必要があります。
- S-VLANのポリシーの適用は、ポリシーがすでに加入者にインストールされている場合、拒否されます。
- S-VLAN ポリシーの削除は、その S-VLAN の下に加入者ポリシーが存在する場合、拒否されます。

手順の概要

1. **configure**
2. **interface** *type*
3. **service-policy output** *name* **subscriber-parent**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.1`	Bundle-Etherアクセスインターフェイスで加入者を設定します。
ステップ 3	**service-policy output** *name* **subscriber-parent** 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy output svlan subscriber-parent`	subscriber-parent キーワードで S-VLAN ポリシーを設定します。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

S-VLAN でのポリシーの設定：例

```
configure
interface Bundle-Ether1.1
service-policy output svlan_pmap subscriber-parent
end
!
```

アクセス インターフェイスでの VLAN ポリシーの設定

アクセス インターフェイスで入力および出力 QoS VLAN ポリシーを適用するには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** *type*
3. **service-policy input** *service-policy-name*
4. **service-policy output** *service-policy-name* [**subscriber-parent**]
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type* 例： `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether18.203`	Bundle-Ether アクセス インターフェイスで加入者を設定します。
ステップ 3	**service-policy input** *service-policy-name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy input mark`	アクセスインターフェイスで入力 VLAN QoS ポリシーを設定します。

ステップ 4	**service-policy output** *service-policy-name* [**subscriber-parent**]
	例：
	`RP/0/RSP0/CPU0:router(config-if)# service-policy output metering`
	または
	`RP/0/RSP0/CPU0:router(config-if)# service-policy output metering subscriber-parent`
		コマンドが **subscriber-parent** キーワードなしで使用される場合、VLAN の出力 QoS ポリシーを設定します。
		コマンドが **subscriber-parent** キーワードとともに使用される場合、S-VLAN の出力 QoS ポリシーを設定します。

ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アクセス インターフェイスでの入力および出力 VLAN ポリシーの設定：例

```
//Applying Ingress and Egress VLAN Policies on an Access Interface

configure
interface Bundle-Ether1.1
service-policy input INGRESS_MARKING_POLICING_POLICY
service-policy output VLAN_POLICY
end
!


//Applying Ingress VLAN Policy and Egress S-VLAN Policies on an Access Interface

configure
interface Bundle-Ether1.2
service-policy input INGRESS_MARKING_POLICING_POLICY
service-policy output S_VLAN_POLICY subscriber-parent
end
!
```

その他の関連資料

ここでは、QoS の実装に関連する参考資料を示します。

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

第 7 章

加入者機能の設定

BNG で設定される加入者機能によって、サービス プロバイダーは、特定のネットワーク リソースの使用制限、司法当局（LEA）への電子機器を使用した情報収集の許可、加入者へのマルチキャスト サービスの提供などの特定の固有機能を導入できます。 この章で説明する加入者機能は、次のとおりです。

過剰なパント フロー トラップ

過剰なパント フロー トラップ機能は、制御パケット トラフィックに割り当てられた共有よりも多く送信するリモート デバイスからの制御パケット トラフィックを識別して緩和しようとします。 リモートデバイスは、加入者デバイス、VLANインターフェイス上のデバイス、または送信元 MAC アドレスで識別されるデバイスの場合があります。

リモートデバイスが制御パケットトラフィックをルータに送信すると、制御パケットは、ルータの CPU を保護するために Local Packet Transport Services（LPTS）キューによってパントされ、ポリシングされます。 1 台のデバイスが過剰なレートの制御パケット トラフィックを送信する場合、ポリサーのキューがいっぱいになり、多くのパケットがドロップされます。 1 台の「不良」

デバイスからのレートが他のデバイスのレートを大幅に超える場合、他のデバイスのほとんどはルータへの制御パケットを取得しません。 過剰なパント フロー トラップ機能は、この状況に対処します。

（注） 過剰なパント フロー トラップ機能がイネーブルではない場合でも、「不良」は他のデバイスのサービスのみに影響を与えます。ルータをダウンさせることはありません。

過剰なパント フロー トラップ機能は、加入者インターフェイスと、L2 および L3 VLAN サブインターフェイスおよびバンドル仮想インターフェイス（BVI）などの非加入者インターフェイスの両方でサポートされます。 パケットでパントのキューをフラッディングする送信元がインターフェイス ハンドルを使用するデバイスの場合、その不良インターフェイスのすべてのパントはペナルティ ポリシングされます。 各プロトコルのデフォルトのペナルティ レートは、10 プロトコル/秒（pps）です。 そうしないと、送信元がインターフェイス ハンドルのないデバイスの場合、この不良からのすべてのパケットがドロップされます。

（注） リリース 4.2.x では、過剰なパント フロー トラップ機能は、加入者インターフェイスでのみ動作する「加入者コントロール プレーン ポリシング（CoPP）」と呼ばれていました。

過剰なパント フロー トラップ機能の動作

過剰なパント フロー トラップ機能は、物理インターフェイス、サブインターフェイス、BVI および加入者インターフェイスから到着する制御パケット トラフィックをモニタします。 インターフェイスは次の 2 つのカテゴリに分割されます。

- 「親」インターフェイス。その下に他のインターフェイスを設定できます。
- 「非親」インターフェイス。その下にインターフェイスを設定できません。

物理インターフェイスは、VLAN サブインターフェイスがあるため、常に親インターフェイスになります。 BVI は、L2 インターフェイスの「親」であるため、常に親インターフェイスになります。 L3 VLAN サブインターフェイスは、親または非親インターフェイスのいずれかになります。 VLAN サブインターフェイスが加入者に対してイネーブルな場合、親インターフェイスになります。それ以外の場合は、非親インターフェイスです。 加入者インターフェイス（IPoE または PPPoE）は、常に非親インターフェイスです。

フローがトラップされると、過剰なパント フロー トラップ機能は、フローの送信元を識別しようとします。 最初に判断することは、フローが発信されたインターフェイスです。 このインターフェイスが「親」インターフェイスでない場合、この機能では、インターフェイスがフローのエンドポイントの送信元で、ペナルティ ポリシングが適用されるものと仮定します。 トラップされたインターフェイスが「親」インターフェイスの場合、すべてのインターフェイスにペナルティを科す（その下のすべてのインターフェイスにペナルティを科す）のではなく、この機能は不正なフローの送信元 MAC アドレスを取得し、親の下の MAC アドレスからすべてのパケットをドロップします。 プラットフォームの制限により、ペナルティのポリサーを MAC アドレスで適用することはできません。したがって、すべてのパケットがドロップされます。

過剰なパント フロー トラップ機能をイネーブルにする方法の詳細については、過剰なパント フロー トラップ処理のイネーブル化, （272 ページ）を参照してください。

（注） 過剰なパント フロー トラップ機能は、すべてのパント トラフィックをモニタします。 初期のモニタリングから特定のインターフェイスを除外することはなく、インターフェイスが過剰フローの送信元の場合、インターフェイスに不良というフラグを付けないようにすることもできません。

不良は、各プロトコルにポリシングされます。 過剰なパント フロー トラップ機能でサポートされるプロトコルは、ブロードキャスト、マルチキャスト、ARP、DHCP、PPP、PPPoE、ICMP、IGMP、L2TP、および IP（多くのタイプの L3 ベース パントや IPv4 と IPv6 の両方を含む）です。各プロトコルには、静的なパント レートとペナルティ レートがあります。 たとえば、リモート デバイスからのすべての ICMP パントの合計は、ルータの CPU に対して 1500 パケット/秒（pps）でポリシングされます。 1 台のリモート デバイスが過剰なレートの ICMP トラフィックを送信してトラップされた場合、その不良からの ICMP トラフィックは 10 pps でポリシングされます。 残りの（非不良）リモート デバイスは、ICMP に静的な 1500 pps のキューを使用し続けます。

（注） インターフェイスをトラップさせるために必要な過剰なレートは、静的なパント レート（ICMP の場合 1500 pps）とは関係ありません。 過剰なレートは、パントされている他の制御パケットの現在の平均レートよりもかなり大きくなります。 過剰なレートは固定レートではなく、現在の全体的なパント パケット アクティビティによって異なります。

不良がトラップされると、不良と特定したプロトコルとは関係なく、すべてのパント プロトコル（ARP、DHCP、PPP など）でペナルティ ポリシングされます。 10 pps のペナルティ レートによって、他のプロトコルは十分に正常に動作できます。 ただし、不良が送信元 MAC アドレスによってトラップされると、すべてのパケットはドロップされます。

インターフェイスはトラップされると、しばらく「ペナルティ ボックス」に置かれます（デフォルトでは 15 分）。 ペナルティのタイムアウトの最後に、ペナルティ ポリシングから削除（またはドロップ）されます。 リモート デバイスからの制御パケット トラフィックのレートが依然として過剰な場合、インターフェイスは再度トラップされます。

制約事項

次の制約事項が、過剰なパント フロー トラップ機能の実装に適用されます。

- この機能は、SIP-700 ラインカードおよび ASR 9000 イーサネット ラインカードのインターフェイスをサポートしません。
- この機能は、非決定論的です。 場合によっては、過剰なパント フロー トラップ機能は、誤検出することがあります。つまり、正規のパント トラフィックを送信しているインターフェイスをトラップすることがあります。
- 過剰なパント フロー トラップ機能は、さまざまなフローの相対レートに基づいてフローをトラップします。したがって、動作は、周囲のパント レートによって異なります。 他のフローよりもかなり大きなフローは、不良としてとしてトラップされることがあります。 した

がって、この機能は、多くのフローがあると感度が鈍くなり、フローが少ないと感度が増します。

- 制御パケットトラフィックは、バーストで発生する可能性があります。過剰なパントフロートラップには、短いバーストでのトリガーを防ぐ手段がありますが、長いバーストでは誤検出トラップをトリガーする可能性があります。

過剰なパント フロー トラップ処理のイネーブル化

加入者および非加入者の両方のインターフェイスで過剰なパントフロートラップ機能をイネーブルにするには、次の作業を実行します。この作業によって、プロトコルのペナルティポリシングレートとペナルティ タイムアウトも設定できます。

手順の概要

1. **configure**
2. **lpts punt excessive-flow-trap subscriber-interfaces**
3. **lpts punt excessive-flow-trap non-subscriber-interfaces**
4. **lpts punt excessive-flow-trap penalty-rate** *protocol penalty_policer_rate*
5. **lpts punt excessive-flow-trap penalty-timeout** *protocol time*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**lpts punt excessive-flow-trap subscriber-interfaces** 例： `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap subscriber-interfaces`	加入者インターフェイスで過剰なパント フロー トラップ機能をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**lpts punt excessive-flow-trap non-subscriber-interfaces** 例： `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap non-subscriber-interfaces`	非加入者インターフェイスで過剰なパント フロー トラップ機能をイネーブルにします。 （注） ステップ 2 とステップ 3 の両方の設定が適用されると、過剰なパント フロー トラップ機能はすべてのインターフェイスでイネーブルになります。
ステップ 4	**lpts punt excessive-flow-trap penalty-rate** *protocol penalty_policer_rate* 例： `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap penalty-rate icmp 10`	プロトコルのペナルティ ポリシング レートを設定します。ペナルティ ポリサー レートはパケット/秒（pps）単位で、その範囲は 2 ～ 100 です。 （注） プロトコルのペナルティ ポリシング レートは、ポリサー レート プロファイルを消費します。
ステップ 5	**lpts punt excessive-flow-trap penalty-timeout** *protocol time* 例： `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap penalty-timeout igmp 10`	トラップされたインターフェイスがペナルティ ボックスに置かれる期間である、プロトコルのペナルティ タイムアウト値を設定します。ペナルティ タイムアウト値は分単位で、その範囲は 1 ～ 1000 です。デフォルトのペナルティ タイムアウト値は、15 分です。

ステップ 6	次のいずれかのコマンドを使用します。
	
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

過剰なパント フロー トラップ処理のイネーブル化：例

次に、デフォルトのペナルティ タイムアウト（15 分）を使用し、PPP および PPPoE プロトコルのペナルティ レートを 20 pps に設定して、加入者インターフェイスで過剰なパント フロー トラップをイネーブルにする例を示します。

```
configure
lpts punt excessive-flow-trap subscriber-interfaces
lpts punt excessive-flow-trap penalty-rate ppp 20
lpts punt excessive-flow-trap penalty-rate pppoe 20
end
!!
```

次に、デフォルトのペナルティ レート（10 pps）を使用し、ARP のペナルティ タイムアウトを 2 分に設定して、非加入者インターフェイスで過剰なパントフロートラップをイネーブルにする例を示します。

```
configure
lpts punt excessive-flow-trap non-subscriber-interfaces
lpts punt excessive-flow-trap penalty-timeout arp 2
end
!!
```

アクセスコントロールリストおよびアクセスコントロール リストベース転送

アクセス コントロール リスト（ACL）は、加入者のアクセス権を定義するために使用されます。また、コンテンツのフィルタリング、さまざまなネットワーク リソースへのアクセスのブロックなどにも使用されます。

特定のサービスプロバイダーは、ルーティングプロトコルによって計算されたパスを使用する代わりに、特定のパスを介してルーティングされる特定のトラフィックをルーティングする必要があります。たとえば、サービスプロバイダーは、音声トラフィックは特定の高価なルートを通過する一方で、データ トラフィックは通常のルーティングパスを使用することを必要とする場合があります。 これは、宛先にパケットを転送するために使用される ACL 設定でネクスト ホップ アドレスを指定することによって実現されます。 パケット転送のために ACL を使用するこの機能は、ACL ベース転送（ABF）と呼ばれます。

（注） セキュリティ ACL および ABF は、PPPoE PTA セッションにのみ適用できます。

ACL は CLI または XML によって定義されます。ただし、動的なテンプレート、または RADIUS からの VSA によって加入者セッションに適用できます。 ABF の導入（ACL の使用）には、次の段階があります。

- ACL の定義については、アクセスコントロール リストの設定, （275 ページ）を参照してください。
- アクセスインターフェイスに ACL を適用するには、ACL のアクティブ化, （277 ページ）を参照してください。

アクセスコントロール リストの設定

アクセス コントロール リストを作成するには、次の作業を実行します。 たとえば、このアクセス リストは ABF を導入するために作成されます。したがって、ネクスト ホップ アドレスを定義します。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**} **access-list** *access-list-name*
3. *sequence-number* **permit tcp any any**
4. *sequence-number* **permit** {**ipv4** | **ipv6**} **host** *source_address* **nexthop** *source_address destination_address*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	{**ipv4** | **ipv6**} **access-list** *access-list-name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list foo_in`
	または
	
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list foo_in`
		アクセスリストを設定します。

	コマンドまたはアクション	目的
ステップ 3	*sequence-number* **permit tcp any any** 例： `RP/0/RSP0/CPU0:router(config)# 10 permit tcp any any`	TCP トラフィックにアクセス コントロール リストのルールを入力します。

ステップ 4	*sequence-number* **permit** {**ipv4** | **ipv6**} **host** *source_address* **nexthop** *source_address destination_address*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# 10 permit ipv4 host 9.8.8.9 nexthop 6.6.6.6 7.7.7.7`
	または
	`RP/0/RSP0/CPU0:router(config)# 10 permit ipv6 host 192:2:1:9 nexthop 192:2:6:8`
		送信元 IP アドレスから宛先 IP アドレスに IPv4 プロトコルで転送するパケットを指定します。
		（注） ステップ 1 から 4 を繰り返して、foo_out アクセス リストを設定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アクセスコントロール リストの設定：例

```
//For IPv4
configure
ipv4 access-list foo_in
10 permit tcp any any
10 permit ipv4 host 9.8.8.9 nexthop 6.6.6.6 7.7.7.7
!
!
end

//For IPv6
configure
ipv6 access-list foo_in
10 permit tcp any any
10 permit ipv4 host 192:2:1:9 nexthop 192:2:6:8
!
!
end
```

ACL のアクティブ化

アクセスコントロール リストをアクティブ化するために使用される動的なテンプレートを定義するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type**{**ipsubscriber** |**ppp** |**service**} *dynamic-template-name*
4. {**ipv4** | **ipv6**} **access-group** *access-list-name* **ingress**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**type**{**ipsubscriber** |**ppp** |**service**} *dynamic-template-name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service foo`	サービスの動的なテンプレート タイプを作成します。

ステップ 4	{**ipv4** | **ipv6**} **access-group** *access-list-name* **ingress**
	
	例：
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 access-group foo_in ingress`
	または
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 access-group foo_in ingress`
		着信パケットに対してアクセスコントロールを指定します。
		（注） 同様に、foo_out と呼ばれる発信パケットに対して別のアクセスグループを作成します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的

ACL のアクティブ化：例

```
//For IPv4
configure
dynamic-template
type service foo
ipv4 access-group foo_in ingress
!
!
end

//For IPv6
configure
dynamic-template
type service foo
ipv6 access-group foo_in ingress
!
!
end
```

合法的傍受のサポート

合法的傍受によって、司法当局（LEA）は、司法命令または行政命令で許可された電子機器を使用した情報収集を実行できます。 ますます多くの法律が採択され、規制が施行されるのに伴い、サービス プロバイダー（SP）やインターネット サービス プロバイダー（ISP）は、許可された電子監視を明示的にサポートするネットワークを実装する必要性に迫られています。 合法的傍受の指令に従う SP または ISP の種類は、国によって大きく異なります。 米国の合法的傍受のコンプライアンスは、Communications Assistance for Law Enforcement Act（CALEA）によって指定されています。

Cisco ASR 9000 シリーズ ルータでは、Cisco Service Independent Intercept（SII）アーキテクチャと PacketCable™ をサポートします。[1] 合法的傍受アーキテクチャ。 合法的傍受コンポーネントだけでは、該当する規制に準拠できませんが、SP および ISP が合法的傍受準拠のネットワークを構築するために使用可能なツールを提供します。

BNG は、加入者のセッション単位の合法的傍受および RADIUS ベースの合法的傍受をサポートします。 セッション単位および RADIUS ベースの両方の合法的傍受が、BNG の IPoE、PPPoE、および PPPoE LAC 加入者セッションで実行されます。

注意 このガイドは、合法的傍受の実装の法的義務に対応するものではありません。 サービス プロバイダーは、ネットワークが適切な合法的傍受の法令および規制に従うことを保障する責任があります。 義務を決定するために法的助言を求めることが推奨されます。

[1] PacketCable™ アーキテクチャは PacketCable™ 仕様を使用してデバイスの相互運用性と製品のコンプライアンスの問題を処理します。

（注） 合法的傍受関連のルータの設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Implementing Lawful Intercept*」の章を参照してください。

セッション単位の合法的傍受

入力方向と出力方向の両方で、複製されたストリームをメディエーション デバイスに送信する、指定された加入者インターフェイス上のすべてのレイヤ 2 またはレイヤ 3 トラフィックの合法的傍受は、セッション単位の合法的傍受と呼ばれます。 この合法的傍受は、シスコが定義する MIB を使用して、IPv4、IPv6、およびマルチキャスト トラフィックの傍受を実装します。 デフォルトでは、SNMP ベースの合法的傍受機能は、Cisco ASR 9000 シリーズ ルータでイネーブルになっており、タップを設定できます。 SNMP ベースの合法的傍受のディセーブル化の詳細については、SNMP ベースの合法的傍受のディセーブル化, （281 ページ）を参照してください。

加入者セッションは、アカウント セッション ID によって識別されます。この ID は、トラフィックが傍受される加入者ユーザに指定された加入者インターフェイスを識別するキーとして機能します。

合法的傍受は通常、SII アーキテクチャまたは PacketCable™ 仕様を使用して実装できます。 SNMP ベースの合法的傍受の Cisco IOS-XR 実装は、サービスに依存しない傍受（SII）のアーキテクチャに基づいています。 SNMPv3 は、データ送信元を認証し、Cisco ASR 9000 シリーズ ルータからメディエーションデバイスへの接続が安全であることを保障します。 これにより、許可されていないパーティが傍受のターゲットを偽造できないようにします。

（注） 合法的傍受を実装するには、SNMP サーバの機能を理解する必要があります。 このため、『*Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*』の「*Implementing SNMP*」モジュールに説明されている情報をよく確認してください。

合法的傍受は、明示的にディセーブルにする必要があります。 これは、プロビジョニングされたルータで自動的にイネーブルになります。 ただし、進行中のアクティブなタップがある場合、タップは削除されるため、LI をディセーブルにしないてください。

管理プレーンは、SNMPv3 をイネーブルにするように設定される必要があります。 コマンドがルータ上のインターフェイス（できれば、ループバック）に機能するように、管理プレーンによる SNMP コマンドの受け入れを可能にします。 これにより、メディエーション デバイス（MD）が物理インターフェイスと通信できるようになります。 管理プレーン保護機能の詳細については、インバンド管理プレーン保護機能の設定, （282 ページ）を参照してください。 メディエーション デバイスのイネーブル化の詳細については、VoIP およびデータ セッションを傍受するためのメディエーション デバイスのイネーブル化, （282 ページ）を参照してください。

合法的傍受 MIB

コレクタとしても知られている外部メディエーション デバイスは、IP-TAP-MIB を使用する IPv4 または IPv6 アドレス ベースのタップを作成できます。 SNMPv3 プロトコルは、メディエーション デバイス（CISCO-TAP2-MIB で定義）、およびタップ（CISCO-USER-CONNECTION-TAP-MIB で定義）のプロビジョニングに使用されます。 Cisco ASR 9000 シリーズ ルータは、SNMP および RADIUS の両方を含む合計 511 の同時タップをサポートします。

合法的傍受は、傍受に次の MIB を使用します。

- CISCO-TAP2-MIB：合法的傍受処理に使用されます。 Cisco ASR 9000 シリーズ ルータで合法的傍受を制御する SNMP 管理オブジェクトが含まれます。 メディエーション デバイスは、Cisco ASR 9000 シリーズ ルータを通してトラフィックを送信するターゲットに対して合法的傍受を設定および実行するために MIB を使用します。 CISCO-TAP2-MIB は、SII 機能をサポートし、メディエーションデバイスおよび汎用タップのプロビジョニングを定義します。これは、メディエーション デバイスのテーブルとストリーム テーブルから主に構成されています。 メディエーション デバイス テーブルには、Cisco ASR 9000 シリーズ ルータが通信するメディエーションデバイスに関する情報が含まれます。たとえば、デバイスアドレス、傍受したトラフィックを送信するインターフェイス、および傍受したトラフィックを送信するために使用するプロトコルです。 ストリーム テーブルには、MD のテーブル エントリによってプロビジョニングされる汎用タップのリストが含まれます。
- CISCO-USER-CONNECTION-TAP-MIB：個々の加入者のトラフィックを傍受するために使用されます。 MIB には、Cisco ASR 9000 シリーズ ルータの個々のユーザ接続に傍受を設定して実行するための SNMP 管理オブジェクトが含まれます。 この MIB には、一意のセッション ID でそれぞれ識別される、ユーザ接続に関する情報が含まれます。
  CISCO-USER-CONNECTION-TAP-MIB は CISCO-TAP2-MIB を設定しないと設定できません。

（注） SNMP タップと RADIUS タップを同時に設定することはできません。 また、同一のセッションを同時に複数回傍受することもできません。

SNMP ベースの合法的傍受のディセーブル化

合法的傍受は、Cisco ASR 9000 シリーズ ルータではデフォルトでイネーブルになっています。

- 合法的傍受をディセーブルにするには、グローバル コンフィギュレーション モードで **lawful-intercept disable** コマンドを入力します。
- この機能を再度イネーブルにするには、このコマンドの **no** 形式を使用します。

SNMP ベースの合法的傍受のディセーブル化：例

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# lawful-intercept disable
```

（注） すべての SNMP ベースのタップは、合法的傍受がディセーブルのときはドロップします。

インバンド管理プレーン保護機能の設定

別のプロトコルを使用するように MPP を設定していない場合、合法的傍受用途で SNMP サーバにメディエーション デバイスとの通信を許可するように MPP 機能も設定されていないことを確認します。 このような場合、指定したインターフェイスまたはすべてのインターフェイスを使用して SNMP コマンドがルータで許可されるように、MPP が明確にインバンド インターフェイスとして設定される必要があります。

（注） Cisco IOS から Cisco IOS XR ソフトウェアに最近移行し、MPP を所定のプロトコルに設定した場合でも、このタスクを必ず実行します。

合法的傍受では、多くの場合にループバック インターフェイスが SNMP メッセージに適しています。 このインターフェイスタイプを選択した場合、インバンド管理設定にこれを含める必要があります。

VoIPおよびデータ セッションを傍受するためのメディエーションデバイスのイネーブル化

次の SNMP サーバ設定作業では、MD による VoIP またはデータ セッションの傍受を許可することで、Cisco IOS XR ソフトウェアを実行しているルータ上で Cisco SII 機能をイネーブルにします。

手順の概要

1. **configure**
2. **snmp-server view** *view-name* **ciscoTap2MIB included**
3. **snmp-server view** *view-name* **ciscoUserConnectionTapMIB included**
4. **snmp-server group** *group-name* **v3auth read** *view-name* **write** *view-name* **notify** *view-name*
5. **snmp-server host** *ip-address* **traps version 3 auth** *username* **udp-port** *port-number*
6. **snmp-server user** *mduser-id groupname* **v3 auth md5** md-*password*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
8. show snmp users
9. show snmp group
10. show snmp view

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**snmp-server view** *view-name* **ciscoTap2MIB included** 例： `RP/0//CPU0:router(config)# snmp-server view TapName ciscoTap2MIB included`	ビュー レコードを作成または変更し、ビューに CISCO-TAP2-MIB ファミリを含めます。 合法的傍受を制御する CISCO-TAP2-MIB 内の SNMP 管理オブジェクトが含まれます。 ルータを通してトラフィックを送信するターゲットで合法的傍受を設定および実行するため、メディエーション デバイスによってこの MIB が使用されます。
ステップ 3	**snmp-server view** *view-name* **ciscoUserConnectionTapMIB included** 例： `RP/0//CPU0:router(config)# snmp-server view TapName ciscoUserConnectionTapMIB included`	ビュー レコードを作成または変更し、ユーザ接続用のシスコの傍受機能を管理するために CISCO-USER-CONNECTION-TAP-MIB ファミリを含めます。 この MIB は、CISCO-TAP2-MIB とともに、ユーザ トラフィックを傍受およびフィルタリングするために使用されます。
ステップ 4	**snmp-server group** *group-name* **v3auth read** *view-name* **write** *view-name* **notify** *view-name* 例： `RP/0//CPU0:router(config)# snmp-server group TapGroup v3 auth read TapView write TapView notify TapView`	SNMP ビューに SNMP ユーザをマッピングする新しい SNMP グループを設定します。 このグループは SNMP ビューの読み取り、書き込み、および通知権限を持っています。
ステップ 5	**snmp-server host** *ip-address* **traps version 3 auth** *username* **udp-port** *port-number* 例： `RP/0//CPU0:router(config)# snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555`	SNMP トラップ通知、使用する SNMP のバージョン、通知のセキュリティ レベル、通知の受信者（ホスト）を指定します。
ステップ 6	**snmp-server user** *mduser-id groupname* **v3 auth md5** md-*password* 例： `RP/0//CPU0:router(config)# snmp-server mduser-id TapGroup v3 auth md5 mdpassword`	MD パスワードと関連付ける v3 セキュリティ モデルと HMAC MD5 アルゴリズムを使用して、MD ユーザが SNMP グループに属するように設定します。 • *mduser-id* および *mdpassword* は MD に設定されている値と一致している必要があります。 あるいは、これらの値はルータで使用されている値と一致している必要があります。

	コマンドまたはアクション	目的
		• SNMPv3 セキュリティの最低基準を満たすには、パスワードの長さは 8 文字以上である必要があります。 • 最低の合法的傍受のセキュリティ レベルは auth です。noauth オプションは noAuthnoPriv セキュリティ レベルを示すため、機能しません。 合法的傍受のセキュリティ レベルは、MDS のレベルと一致している必要があります。 • ルータでは MD5 以外を選ぶこともできますが、MD 値は一致している必要があります。 ほとんどの MD では、MD5 がデフォルトになっているか、MD 5 のみをサポートしています。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 8	show snmp users 例： `RP/0//CPU0:router# show snmp users`	SNMP ユーザ テーブルの各 SNMP ユーザ名に関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 9	show snmp group 例： `RP/0//CPU0:router# show snmp group`	ネットワークの各 SNMP グループに関する情報を表示します。
ステップ 10	show snmp view 例： `RP/0//CPU0:router# show snmp view`	関連する MIB ビュー ファミリー名、ストレージ タイプ、ステータスなど、設定されたビューに関する情報を表示します。

VoIP およびデータ セッションを傍受するためのメディエーション デバイスのイネーブル化：例

```
configure
snmp-server view TapName ciscoTap2MIB included
snmp-server view TapName ciscoUserConnectionTapMIB included
snmp-server group TapGroup v3 auth read TapView write TapView notify TapView
snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555
snmp-server mduser-id TapGroup v3 auth md5 mdpassword
end
!
!
```

RADIUS ベースの合法的傍受

RADIUS ベースの合法的傍受機能は、BNG の加入者トラフィックの傍受に RADIUS 属性を使用して機能を提供します。 これは、IP アドレスがセッションに割り当てられるまで、SNMP ベースの方法はタップされるセッションを回避するため、SNMP ユーザ接続の MIB 上で推奨される方法です。 RADIUS ベースの LI メカニズムでは、セッションが確立されるとすぐにタップが可能になります。

RADIUS ベースの合法的傍受ソリューションによって、RADIUS サーバからネットワーク アクセス サーバ（NAS）またはレイヤ 2 トンネル プロトコル アクセス コンセントレータ（LAC）に（Access-Accept パケットまたは許可変更（CoA）要求パケットを介して）傍受要求を送信できます。 PPP または L2TP セッションとやり取りされるすべてのトラフィック データは、メディエーション デバイスに渡されます。 RADIUS ベースの合法的傍受ソリューションのもう 1 つの利点は、すべてのターゲットのトラフィックを同時に傍受できる Access-Accept パケットを使用してタップを設定することです。

RADIUS ベースの合法的傍受機能は、次のモードのタップ開始サポートを提供しています。

- 新しいセッションに対する Access-Accept ベースの合法的傍受
- 既存のセッションに対する CoA ベースの合法的傍受

（注） デフォルトでは、RADIUSベースの合法的傍受機能は、イネーブルになっていません。RADIUSベースの合法的傍受のイネーブル化の詳細については、RADIUSベースの合法的傍受のイネーブル化，（286 ページ）を参照してください。

RADIUS ベースの合法的傍受のイネーブル化

RADIUS ベースの合法的傍受機能をイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **aaa intercept**
3. **aaa server radius dynamic-author**
4. **port** *port_number*
5. **server-key** [**0**|**7**] *word*
6. **client** *hostname*{ **vrf** *vrf_name* | **server-key** [**0**|**7**] *word* }
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aaa intercept** 例： `RP/0/RSP0/CPU0:router(config)# aaa intercept`	RADIUS ベースの合法的傍受機能をイネーブルにします。 （注） AAA の傍受をディセーブルにすると、すべての RADIUS ベースのタップが Cisco ASR 9000 シリーズ ルータから削除されます。
ステップ 3	**aaa server radius dynamic-author** 例： `RP/0/RSP0/CPU0:router(config)# aaa server radius dynamic-author`	合法的傍受を AAA サーバとして設定し、動的許可ローカル サーバ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**port** *port_number* 例： `RP/0/RSP0/CPU0:router(config-Dynamic Author)# port 1600`	RADIUS サーバ ポートを指定します。 デフォルト ポート番号は、1700 です。
ステップ 5	**server-key** [**0**|**7**] *word* 例： `RP/0/RSP0/CPU0:router(config-Dynamic Author)# server-key cisco`	RADIUS クライアントと共有される暗号キーを指定します。
ステップ 6	**client** *hostname*{ **vrf** *vrf_name* | **server-key** [**0**|**7**] *word* } 例： `RP/0/RSP0/CPU0:router(config-Dynamic Author)# client 3.0.0.28 vrf default server-key cisco`	AAA サーバが通信するクライアントを指定します。 （注） グローバル モード、およびクライアント タイプのキー単位としてサーバ キーを設定できます。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

RADIUS ベースの合法的傍受のイネーブル化：例

```
configure
aaa intercept
aaa server radius dynamic-author
port 1600
server-key cisco
client 3.0.0.28 vrf default server-key cisco
end
!
!
```

次の作業

次の属性は、RADIUS ベースの合法的傍受を設定するために、ユーザ プロファイルに含まれている必要があります。

```
xyz_user1@domain.com Password == "cisco"
    Cisco-avpair = "md-ip-addr=192.1.1.4",
    Cisco-avpair += "md-port=203",
    Cisco-avpair += "md-dscp=3",
    Cisco-avpair += "intercept-id=abcd0003",
    Cisco-avpair += "li-action=1"
```

TCP MSS 調整

TCP MSS 調整機能によって、Cisco ASR 9000 シリーズ ルータを通過する一時的なパケットの最大セグメント サイズ（MSS）を設定できます。

PPPoE または L2TP のケースでは、TCP セッションを開始するクライアントが気づかないことがある追加ヘッダーがパケットに追加されます。 これにより、追加されたヘッダーが原因でパケット サイズが最大伝送単位（MTU）を超えた場合に、損失パケット、破損した送信、またはフラグメンテーションが生じる可能性があります。

TCP MSS 調整機能がどのように機能するのかを示す例を次に示します。

図 *9*：*TCP MSS* 調整の例

この例では、HTTP クライアントは、1300（MTU）- 20 TCP - 20 IP ヘッダー = 1260 の MSS 値を信号で伝える TCP 同期（SYN）パケットを HTTP サーバに送信します。 これを受信すると、HTTP サーバは SYN ACK メッセージで通知します。 HTTP クライアントは、単一の確認応答で TCP セッションを確認し、TCP チャネルを開きます。

（注） これは、PPPoE または L2TP なしのサンプル シナリオです。

HTTP サーバは大きなファイルを選択すると、それを 1460 バイトのチャンクに分割します（現時点で HTTP ヘッダーは存在しないと想定します）。 HTTP サーバがパケットを送信すると、最初の Cisco ASR 9000 シリーズ ルータ（右）は、MTU がクライアントに対して 576 ダウンストリームであることを検出し、1300 バイトのパケットをフラグメント化するよう要求します。

サーバが DF（「フラグメント化しない」）ビットを設定すると、パケットはドロップされます。また、パケットに DF ビットが設定されていない場合、パケットはフラグメント化され、パケットを再構成するようクライアントに要求します。デジタル加入者線（DSL）または Fibre-to-the-Home（FTTH）のようなアクセスでは、CPE がセキュリティ メカニズムとして着信フラグメントをブロックし、この送信が失われることがあります。

一般的なシナリオでは、ドロップされるパケットがあると、Web ページでイメージを表示するときに、部分的なダウンロード、障害、または遅延が発生します。MSS 調整は、サーバが設定されたサイズよりも大きなパケット（およびヘッダー）を送信しないように、TCP SYN パケットの傍受、MSS オプションの読み取り、および値の調整を行うことによってこのシナリオを克服します。

TCP MSS 値のみが下方調整されることに注意してください。クライアントが設定値よりも小さい MSS 値を要求する場合、実行されるアクションはありません。

PPPoE の場合は余分な 8 バイト、L2TP の場合は余分な 40 バイトが、パケットに追加されます。推奨される MSS 調整値は、PPPoE の場合は 1452、L2TP の場合は 1420 で、1500 エンドツーエンドの最小 MTU を想定しています。

PTA と L2TP に対して異なる一意のグローバル値がサポートされています。これが一度設定されると、今後すべてのセッションを TCP MSS 調整できます。ただし、すでに確立されたセッションは TCP 調整されません。 グローバル値が変更されると、すべての新しい TCP 加入者セッションは、新しいグローバル値を取得します。

パケットの TCP MSS 値の設定の詳細については、TCP パケットの TCP MSS 値の設定, （290 ページ）を参照してください。

（注） セッションでこの機能をディセーブルにするには、まずグローバル コンフィギュレーションをディセーブルにしてから、セッションを削除し、再作成します。

IPv4 と IPv6 の両方でカプセル化された TCP がサポートされます。

制約事項

次の制約事項が、TCP MSS 調整に適用されます。

- MSS は TCP 固有のため、TCP MSS 調整機能は、（中継）TCP パケットのみに適用でき、UDP パケットには影響しません。
- TCP MSS 調整の設定は、PPPoE PTA および LAC セッション タイプのみに影響します。 これは、IP セッションまたは非 BNG インターフェイスには影響しません。
- MSS オプションは、TCP ヘッダーの最初のオプションである必要があります。
- ルータは、ユーザが TCP/IPV4 パケットを検査するために設定した MSS 値を使用します。TCP/IPv6 パケットを調べると、ルータは、より大きな IPv6 ヘッダーを考慮して、設定され

た MSS 値を自動的に 20 バイトずつ下方調整します。 たとえば、TCP MSS 値が 1450 に設定されている場合、ルータは IPv4 パケットの TCP MSS を 1450 に下方調整し、IPv6 パケットでは 1430 に下方調整します。

TCP パケットの TCP MSS 値の設定

TCP パケットの TCP MSS 値を TCP セッションがドロップされるのを防ぐように設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **subscriber**
3. **pta tcp mss-adjust** *max-segment-size*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. **configure**
6. **vpdn**
7. **l2tp tcp-mss-adjust** *max-segment-size*
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**subscriber** 例： `RP/0/RSP0/CPU0:router(config)# subscriber`	加入者コンフィギュレーション モードをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**pta tcp mss-adjust** *max-segment-size* 例： `RP/0/RSP0/CPU0:router(config-subscriber)# pta tcp mss-adjust 1300`	PTA 加入者用の Cisco ASR 9000 シリーズ ルータを介して送信される TCP パケットの MSS 値を設定します。 TCP MSS 調整の最大セグメント サイズの範囲は、1280 ～ 1536（バイト単位）です。 （注） 値は、機能がイネーブルな場合にすべてのセッションに適用される、PTA セッションのグローバル値を表します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 5	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 6	**vpdn** 例： `RP/0/RSP0/CPU0:router(config)# vpdn`	VPDN コンフィギュレーション モードをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 7	**l2tp tcp-mss-adjust** *max-segment-size* 例： `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp tcp-mss-adjust 1300`	LAC 加入者用の Cisco ASR 9000 シリーズ ルータを介して送信される TCP パケットの MSS 値を設定します。 TCP MSS 調整の最大セグメント サイズの範囲は、1280 ～ 1460（バイト単位）です。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

TCP パケットの TCP MSS 値の設定：例

```
//Example for configuring the TCP MSS value of TCP packets for a PPPoE PTA subscriber
session:

configure
subscriber
pta tcp mss-adjust 1280
!!


// Example for configuring the TCP MSS value of TCP packets for a PPPoE LAC subscriber
session:

configure
vpdn
```

```
l2tp tcp-mss-adjust 1460
!!
```

あいまいな VLAN の加入者セッション

あいまいな VLAN によって、単一のアクセスインターフェイスで複数の加入者セッションを作成できます。 結果として、アクセスインターフェイスの拡張性が向上します。 あいまいな VLAN は、VLAN ID の範囲、または個々の VLAN ID のグループのいずれかが指定されている L3 インターフェイスです。 VLAN に各加入者を個々にマッピングする代わりに、あいまいな VLAN 設定ではグループのマッピングを実行します。 複数の加入者は、一意の MAC アドレスがある限り、あいまいな VLAN でマッピングできます。 あいまいな VLAN に作成された加入者セッションは、通常の VLAN に作成されているものと同じで、ポリシーマップ、VRF、QoS、アクセスコントロール リストなどのすべての通常設定をサポートしています。

あいまいな VLAN での IPoE 加入者セッションの作成をイネーブルにするには、あいまいな VLAN での加入者セッションの確立, （293 ページ）を参照してください。

制約事項

あいまいな VLAN は、ユニキャスト クライアントを使用しません。

あいまいな VLAN での加入者セッションの確立

あいまいな VLAN を定義し、そこでの IP 加入者セッションの作成をイネーブルにするには、次の作業を実行します。

（注） あいまいな VLAN に必要な DHCP 固有の設定はありません。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のコマンドのいずれかを使用して、カプセル化されたあいまいな VLAN を設定します。
   - **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* }
   - **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* }
   - **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* }
   - **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
4. **ipv4 | ipv6address** *source-ip-address destination-ip-address*
5. **service-policy type control subscriber** *policy_name*
6. **ipsubscriber ipv4 l2-connected**
7. **initiator dhcp**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/1/0/0.12`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	次のコマンドのいずれかを使用して、カプセル化されたあいまいな VLAN を設定します。 • **encapsulation ambiguous** { **dot1q**	**dot1ad** } {**any**	*vlan-range* } • **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any**	*vlan-range* } • **encapsulation ambiguous dot1q any second-dot1q** { **any**	*vlan-id* }	IEEE 802.1Q VLAN を設定します。 *vlan-range* は、例に示すように、カンマ区切りまたはハイフン区切りの形式、またはその両方の組み合わせで指定されます。

	コマンドまたはアクション	目的

	• **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 14 second-dot1q 100-200`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any second-dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1ad 14 dot1q 100,200,300-400`	

ステップ 4	**ipv4 | ipv6address** *source-ip-address destination-ip-address*
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 2.1.12.1 255.255.255.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv6 address 1:2:3::4 128`
		IPv4 または IPv6 プロトコル アドレスを設定します。

ステップ 5	**service-policy type control subscriber** *policy_name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	指定された PL1 の policy_name でポリシーマップが前に定義された、アクセスインターフェイスにポリシーマップを適用します。
ステップ 6	**ipsubscriber ipv4 l2-connected** 例： `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv4 l2-connected`	L2 接続された IPv4 IP 加入者をイネーブルにします。
ステップ 7	**initiator dhcp** 例： `RP/0/RSP0/CPU0:router(config-if)# initiator dhcp`	IP 加入者の発信側 DHCP をイネーブルにします。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

あいまいな VLAN での加入者セッションの確立：例

```
configure
interface Bundle-Ether100.10
encapsulation ambiguous dot1q 14 second-dot1q any
ipv4 address 2.1.12.12 55.255.255.0
service-policy type control subscriber PL1
ipsubscriber ipv4 l2-connected
!
!
end
```

uRPF

ユニキャスト リバース パス転送（uRPF）は、加入者インターフェイスで受信されるパケットが有効な加入者から送信されているかどうかを確認する BNG の機能です。 uRPF は、L3 サービスを使用する加入者にのみ適用されます。

PPPoE加入者の場合、uRPFチェックで、着信パケットの送信元アドレスが加入者に関連付けられたアドレス セットに一致することを確認します。 加入者アドレスは、IPCP の割り当てられたアドレス、またはRADIUS を介してフレーム化され、ルーティングされ、割り当てられたアドレスです。 PPPoE 加入者は、セッション ID と VLAN キーで識別されます。 BNG は、着信パケットの送信元 IP アドレスが予期されたセッション ID と VLAN キーに一致することを確認する uRPF チェックを実行します。

IPoE加入者の場合、加入者アドレスはDHCPによって割り当てられたものです。 IPoE加入者は、着信 MAC アドレスで識別されます。 uRPF チェックでは、送信元 IP アドレスが DHCP によって送信元 MAC アドレスに割り当てられたものであることを確認します。

uRPF は、IPv4 および IPv6 の両方の加入者でサポートされ、動的なテンプレートを使用できます。uRPF をイネーブルにする動的なテンプレートを定義するには、IPv4 または IPv6 加入者セッションの動的なテンプレートの作成, （96 ページ）を参照してください。

マルチキャスト サービス

マルチキャスト サービスにより、複数の加入者を 1 つの送信元からの単一送信の受信者にすることができます。たとえば、リアルタイム音声およびビデオ会議では、マルチキャスト サービスが有効活用されます。 BNG の PPPoE インターフェイスで適用されるマルチキャスト機能は、次のとおりです。

マルチキャストの共存

BNG で、マルチキャスト サービスは、通常のユニキャスト サービスと共存します。 BNG のマルチキャスト機能は、Cisco ASR 9000 シリーズ ルータでサポートされている既存の L3 マルチキャスト機能と同じです。 BNG では、マルチキャストはトランク インターフェイス、および物理インターフェイスおよびバンドルで作成された VLAN でイネーブルになります。 マルチキャストの共存は PPPoE PTA 加入者セッションに対して機能します。 ASR9k でのマルチキャストの実装の詳細については、『Implementing Layer 3 Multicast Routing on Cisco ASR 9000 Series Routers』を参照してください。

BNG でマルチキャスト機能をイネーブルにするには、VRF のアドレス ファミリのイネーブル化, （297 ページ）を参照してください。

VRF のアドレス ファミリのイネーブル化

必要なアドレス ファミリのマルチキャスト機能をイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **multicast-routing**
3. **vrf** *vrf_name*
4. **address-family ipv4**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**multicast-routing** 例： `RP/0/RSP0/CPU0:router(config)# multicast routing`	マルチキャストルーティングを設定します。
ステップ 3	**vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:router(config)# vrf vrf1`	VRF 名を設定します。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config)# address-family ipv4`	IPv4 アドレス ファミリのマルチキャスト機能をイネーブルにします。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

VRF のアドレス ファミリのイネーブル化：例

```
multicast-routing
vrf vrf1
address-family ipv4
 !
!
end
```

マルチキャスト レプリケーション

BNGは、PPPoEインターフェイスのマルチキャストパケットのレプリケーションをサポートします。 また、加入者インターフェイス上のマルチキャスト転送、マルチキャスト IP ビデオ コンテンツの送信もサポートします。マルチキャスト レプリケーションが加入者でイネーブルになっている場合、BNG はその加入者に対して IGMP 統計情報の収集を実行し、その情報をエクスポートできます。 マルチキャスト レプリケーションは、パッシブ モードで設定された加入者インターフェイスでサポートされます。

HQoS 相関

階層型 Quality of Service（HQoS）の相関機能は、各加入者の PPPoE セッションで受信した IGMP レポートで加入者のマルチキャスト帯域幅の使用状況をモニタし、マルチキャスト トラフィックに十分な帯域幅を残すようにユニキャスト帯域幅の使用を制限します。 これは、マルチキャストおよびユニキャスト トラフィックが異なるデバイスによってラスト マイル リンク上で転送される場合に、マルチキャスト トラフィックとユニキャスト トラフィックがラスト マイルの加入者への同一物理リンクを共有するときに便利です。 この機能は、加入者にユニキャスト トラフィックを転送する BNG で設定されます。 受信された IGMP レポートに基づいて、BNG は、PPPoE セッションでユニキャスト QoS シェーパーを通知し、ユニキャスト トラフィック フローに許可された帯域幅制限を変更します。 この HQoS 相関機能を使用して、サービス プロバイダーは、集中的なユニキャスト トラフィックから PPPoE 加入者へのマルチキャスト トラフィックを保護できます。 マルチキャスト フローの帯域幅プロファイルは、BNG で設定される必要があります。

帯域幅プロファイルを定義するには、最小ユニキャスト帯域幅の設定, （300 ページ）を参照してください。

マルチキャスト HQoS のモードを指定するには、マルチキャスト HQOS 相関モードまたはパッシブ モードの設定, （302 ページ）を参照してください。

最小ユニキャスト帯域幅の設定

ユニキャスト トラフィックがオーバーサブスクライブ マルチキャスト トラフィックによって完全に切断されないように、最小ユニキャスト帯域幅を設定できます。 QoS を使用して加入者に対して保証される最小ユニキャスト帯域幅を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type** [**ppp**|**ip-subscriber**|**service**]*name*
4. **qos output minimum-bandwidth** *range*
5. **exit**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 3	**type** [**ppp**|**ip-subscriber**|**service**]*name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp p1`	. 適用する必要がある動的なテンプレートのタイプを指定します。3つの使用可能なタイプは、次のとおりです。 • PPP • IP 加入者 • サービス

	コマンドまたはアクション	目的
ステップ 4	**qos output minimum-bandwidth** *range* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# qos output minimum-bandwidth 10`	加入者に対して保証される最低帯域幅を kBps 単位で設定します。 範囲は 1 ～ 4294967295 です。
ステップ 5	**exit** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# exit`	現在のモードを終了します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

最小帯域幅の設定：例

```
configure
dynamic-template
type ppp p1
service-policy output pmap
multicast ipv4 qos-correlation
qos output minimum-bandwidth 10
end
```

マルチキャスト HQOS 相関モードまたはパッシブ モードの設定

HQoS 相関モードまたはパッシブ モードでマルチキャスト設定し、PPPoE インターフェイス上でマルチキャスト レプリケーションをイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type ppp** *dynamic-template name*
4. **multicast ipv4 <qos-correlation | passive>**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 3	**type ppp** *dynamic-template name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp foo`	PPP タイプ モードを開始し、加入者インターフェイスの IGMP を設定します。

	コマンドまたはアクション	目的
ステップ 4	**multicast ipv4 <qos-correlation | passive>** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# multicast ipv4 qos-correlation`	QoS 相関モード（IGMP-HQOS 相関）またはパッシブ モード（マルチキャスト転送）のいずれかで加入者を設定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

マルチキャスト HQOS 相関モードの設定：例

```
dynamic-template type ppp foo
multicast ipv4 qos-correlation
 !
!
end
```

ユニキャスト QoS シェーパー相関に対する IGMP

ユニキャスト QoS シェーパーの相関機能は、マルチキャスト フローの帯域幅プロファイルを設定し、IGMP メッセージによって各加入者のマルチキャスト帯域幅の使用状況を推測できます。PPPoE 加入者セッションで、加入者が使用するマルチキャスト帯域幅は、最小しきい値に到達するまでユニキャスト QoS シェーパーから差し引かれます。

IGMP QoS シェーパーの設定の詳細については、VRF での IGMP - HQoS 相関機能の設定, （304 ページ）を参照してください。 加入者インターフェイスの IGMP の設定の詳細については、加入者インターフェイスの IGMP パラメータの設定, （308 ページ）を参照してください。

IGMP は、ルートポリシーを使用して、すべてのマルチキャスト フローの絶対レートを分散します。 ユニキャスト QoS シェーパーのルートポリシーの設定の詳細については、ユニキャスト QoS シェーパーのルートポリシーの設定, （306 ページ）を参照してください。

VRF での IGMP - HQoS 相関機能の設定

VRF で IGMP - HQoS 相関機能を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router igmp**
3. **unicast-qos-adjust adjustment-delay** *time*
4. **unicast-qos-adjust download-interval** *time*
5. **unicast-qos-adjust holdoff** *time*
6. **vrf** *vrf-name*
7. **traffic profile** *profile-name*
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**router igmp** 例： `RP/0/RSP0/CPU0:router(config)# router igmp`	IGMP コンフィギュレーション モードのルータ プロセスを開始します。
ステップ 3	**unicast-qos-adjust adjustment-delay** *time* 例： `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust adjustment-delay 1`	加入者のユニキャスト トラフィックに対して、IGMP QoS シェーパーでレートをプログラミングするまで待機する時間を設定します。 待機時間の範囲は 0～10 秒です。
ステップ 4	**unicast-qos-adjust download-interval** *time* 例： `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust download-interval 10`	加入者のユニキャスト トラフィックに対して、IGMP QOS シェーパーにインターフェイスのバッチをダウンロードするまでの時間を設定します。 ダウンロード間隔の範囲は10～500 ミリ秒です。
ステップ 5	**unicast-qos-adjust holdoff** *time* 例： `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust holdoff 5`	QoS が IGMP QoS シェーパーの失効したエントリを消去するまでのホールドオフ時間を設定します。 ホールドオフ時間の範囲は 5 ～ 1800 秒です。
ステップ 6	**vrf** *vrf-name* 例： `RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	VRF コンフィギュレーション モードを開始します。
ステップ 7	**traffic profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# traffic profile routepolicy1`	ルートポリシーが帯域幅プロファイルのマッピングに使用されるように設定します。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

IGMP QOS シェーパーの設定：例

```
configure
router igmp
unicast-qos-adjust adjustment-delay 1
unicast-qos-adjust download-interval 10
unicast-qos-adjust holdoff 5
vrf vrf1
traffic profile routepolicy1
!
!
end
```

ユニキャスト QoS シェーパーのルートポリシーの設定

ユニキャスト QoS シェーパーのルートポリシーを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router igmp**
3. **vrf** *vrf-name*
4. **traffic profile** *profile-name*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
6. **show igmp unicast-qos-adjust statistics**
7. **show igmp unicast-qos-adjust statistics interface** *interface-name*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router igmp** 例： `RP/0/RSP0/CPU0:router(config)# router igmp`	IGMP コンフィギュレーション モードのルータ プロセスを開始します。
ステップ 3	**vrf** *vrf-name* 例： `RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	VRF コンフィギュレーション モードを開始します。
ステップ 4	**traffic profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# traffic profile routepolicy1`	ルートポリシーが帯域幅プロファイルのマッピングに使用されるように設定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show igmp unicast-qos-adjust statistics** 例： `RP/0/RSP0/CPU0:router# show igmp unicast-qos-adjusted statistics`	（任意）調整下のインターフェイス グループの総数、前回の clear コマンドからの稼働時間、およびユニキャスト QoS シェーパーのレート調整コールの総数など、機能の内部統計情報を表示します。
ステップ 7	**show igmp unicast-qos-adjust statistics interface** *interface-name* 例： `RP/0/RSP0/CPU0:router# show igmp unicast-qos-adjusted statistics interface interface1`	（任意）インターフェイス名、調整されたフロー数、調整された合計レート、ユニキャスト QoS シェーパーの初期調整後の稼働時間を表示します。

ユニキャスト QoS シェーパーのルートポリシーの設定：例

```
#Adding a route-policy for profile1

route-policy profile1
if destination in (239.0.0.0/8 le 32) then
set weight 1000
endif
end-policy

# Configuring profile1 for Unicast QoS Shaper
router igmp
vrf vrf1
traffic profile profile1
 !
!
end
```

加入者インターフェイスの IGMP パラメータの設定

加入者インターフェイスの IGMP パラメータを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **dynamic-template**
3. **type ppp** *dynamic-template name*
4. **igmp explicit-tracking**
5. **igmp query-interval** *value*
6. **igmp query-max-response-time** *query-response-value*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template** 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template`	動的テンプレート コンフィギュレーション モードを開始します。
ステップ 3	**type ppp** *dynamic-template name* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp foo`	PPP タイプ モードを開始し、加入者インターフェイスの IGMP を設定します。
ステップ 4	**igmp explicit-tracking** 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp explicit-tracking`	IGMPv3 の明示的ホスト トラッキングをイネーブルにします。
ステップ 5	**igmp query-interval** *value* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp query-interval 60`	IGMP の query-interval を秒単位で設定します。 （注） igmp query-interval の値は、秒単位で 1 ～ 3600 の範囲にする必要があります。16000 PPPoE 加入者以下の場合、推奨値（デフォルト値でもある）は 60 秒です。

	コマンドまたはアクション	目的
ステップ 6	**igmp query-max-response-time** *query-response-value* 例： `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp query-max-response-time 4`	IGMP の query-max-response-time を秒単位で設定します。 （注） igmp query-interval の値は、秒単位で 1 ～12 の範囲です。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

加入者インターフェイスの IGMP の設定：例

```
dynamic-template type ppp foo
igmp explicit-tracking
igmp query-interval 60
igmp query-max-response-time 4
 !
!
end
```

IGMP アカウンティング

インターネット グループ管理プロトコル（IGMP）アカウンティング機能によって、BNG でマルチキャスト グループに加入する加入者または消去される加入者のインスタンスをログ記録する統計情報ファイルを維持できます。 ファイル形式は、次のとおりです。

```
harddisk:/usr/data/igmp/accounting.dat.<Node ID>.<YYMMDD>
```

値は次のとおりです。

- ノード ID は、ファイルを生成するノードの名前です（RP/0/RSP0/CPU0 など）。
- YY は年、MM は月、DD は日付です。

統計情報のファイル名の例は、次のとおりです。

```
harddisk:/usr/data/igmp/accounting.dat.RP_0_RSP0_CPU0.101225
```

統計情報ファイルは、アクティブなルート プロセッサ（RP）に保存されます。 フェールオーバー イベントが発生すると、新しいファイルが新しいアクティブ RP で作成され、アクティブ RP とスタンバイ RP 間でデータをミラーリングする試行は行われません。 したがって、統計情報ファイルはアクティブ RP とスタンバイ RP の両方から取得される必要があります。

デフォルトでは、IGMP アカウンティング機能は、毎日 1 ファイルを追加します。 ディスク領域の枯渇を防ぐために、データを保持するファイル数または日数を指定できます。IGMP アカウンティングの設定, （311 ページ）を参照してください。 指定された期間よりも古いファイルは削除され、データは BNG から廃棄されます。 各ファイルの最大サイズが 250 MB を超えることはできません。

IGMP アカウンティングの設定

IGMP アカウンティングを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router igmp**
3. **accounting** [**max-history**] *days*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. **show igmp interface**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router igmp** 例： `RP/0/RSP0/CPU0:router(config)# router igmp`	IGMP コンフィギュレーション モードのルータ プロセスを開始します。
ステップ 3	**accounting** [**max-history**] *days* 例： `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# accounting max-history 50`	IGMPアカウンティングを設定します。 最大履歴パラメータは任意であり、保持されるファイル数を指定します。この数値は、履歴日数と等しくなります。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 5	**show igmp interface** 例： `RP/0/RSP0/CPU0:router# show igmp interface`	（任意）IGMP インターフェイス情報を表示します。

IGMP アカウンティングの設定：例

```
configure
router igmp
accounting max-history 45
!
!
end
```

DAPS サポート

分散アドレスプールサービス（DAPS）によって、ラインカード（LC）およびルートプロセッサ（RP）で実行される DHCP プロセス間でアドレス プールを共有できます。 DHCP サーバおよび PPPoE加入者は、DAPSに対するクライアントであり、DAPSクライアントと呼ばれます。 RADIUS 属性に属性「プール名」が含まれる場合にのみ、クライアントに IP アドレスを返すために DAPS が使用されます。 加入者の RADIUS 属性に固定アドレスが含まれる場合、クライアントはその IP アドレスを DAPS に送信しません。

DAPS は、RP の DAPS サーバ、および LC の DAPS プロキシの 2 つの形式で実行されます。 RP には、組み込み式の DAPS プロキシ モジュールがあります。 このモデルでは、すべての DAPS クライアントが常に DAPS プロキシと通信します。 DAPS プロキシ インスタンスは、アドレス割り当てや他の要求について RP の中央 DAPS サーバと通信します。 DAPS プロキシは、システム内のすべての LC で動作します。 LC 上で動作する DAPS プロキシは、その LC から複数のクライアントを提供できます（PPP、DHCPv6、IPv6ND など）。 DAPS は、2 つ以上のノードで複数の DAPS クライアントを提供します。 異なる DAPS プロキシ プロセスは、各ノードで実行され、各 DAPS クライアントにローカルに接続します。

DAPS は、プール名による動的な IPv4 および IPv6 アドレス割り当てをサポートします。 IPv4 DAPS の設定の詳細については、IPv4 分散アドレス プール サービスの設定，（314 ページ）を参照してください。 IPv6 の設定プールを作成するには、設定プール サブモードの作成，（315 ページ）を参照してください。

IPv6 コンフィギュレーション サブモードでさまざまな DAPS IPv6 パラメータを設定できます。 IPv6アドレスプールのサブネット番号とマスクを設定できます。詳細については、アドレス プールのサブネット番号およびマスクの設定，（318ページ）を参照してください。IPv6アドレスの範囲などのパラメータを指定できます。 詳細については、「IPv6 アドレスの範囲の指定，（320 ページ）」を参照してください。使用率のしきい値を指定するには、使用率のしきい値の指定，（322 ページ）を参照してください。 1 つのサブネット内で一連のプレフィックスまたはアドレスを指

定するには、サブネット内の一連のアドレスまたはプレフィックスの指定，（326ページ）を参照してください。 プレフィックスの長さも指定できます。 詳細については、「プレフィックスの長さの指定，（324 ページ）」を参照してください。

IPv4 分散アドレス プール サービスの設定

IPv4 分散アドレス プール サービス（DAPS）を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool ipv4** *ipv4-pool-name*
3. **address-range** *first_address second_address*
4. **pool vrf** *vrf-name* **ipv4** *ipv4-pool-name*{**address-range** *address-range*}
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool ipv4** *ipv4-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool ipv4 pool1`	IPv4 プール名を設定します。
ステップ 3	**address-range** *first_address second_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv4)# address-range 1.1.1.1 9.8.9.8`	割り当てのアドレス範囲を設定します。
ステップ 4	**pool vrf** *vrf-name* **ipv4** *ipv4-pool-name*{**address-range** *address-range*} 例： `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv4 pool1 address-range 1.1.1.1 9.8.9.8`	IPv4 プール名を設定します。

	コマンドまたはアクション	目的

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPv4 分散アドレス プール サービスの設定：例

```
pool ipv4 pool1
address-range 1.1.1.1 9.8.9.8
pool vrf vrf1 ipv4 pool1 address-range 1.1.1.1 9.8.9.8
 !
!
end
```

設定プール サブモードの作成

デフォルトの VRF および特定の VRF の IPv6 設定プール サブモードを作成し、イネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool ipv6** *ipv6-pool-name*
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
4. **configure**
5. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool ipv6 pool1`	デフォルトの VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。
ステップ 4	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 5	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 pool1`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

設定プール サブモードの作成：例

```
configure
pool ipv6 pool1 (default vrf)
!
!
configure
pool vrf vrf1 ipv6 pool1 (for a specific vrf)
!
!
end
```

アドレス プールのサブネット番号およびマスクの設定

IPv6 アドレス プールのサブネット番号およびマスクを作成するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. **exclude** *low_ip_address high_ip_address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**prefix-length** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	クライアントに割り当てられたプレフィックスの長さを指定します。プレフィックス長の値の範囲は 1 ～ 128 です。
ステップ 4	**network** *subnet* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	サブネット内の一連のアドレスまたはプレフィックスを指定します。 （注） **prefix-length** コマンドは、**network** コマンドが使用される場合は、必ず設定する必要があります。
ステップ 5	**utilization-mark high** *value* **low** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	プール IPv6 サブモードで使用率のしきい値を指定します。 高い値および低い値は 0 ～ 100 のパーセンテージとして表されます。
ステップ 6	**exclude** *low_ip_address high_ip_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# exclude 1101:1::100 ::`	DAPS がクライアントに割り当てることはできない IPv6 アドレスまたはプレフィックスの範囲を指定します。 高い値および低い値は 0 ～ 100 のパーセンテージとして表されます。 （注） プール内では複数の exclude コマンドが許可されます。1 つのアドレスを除外するには、<*high_ip_address*> を省略できます。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アドレス プールのサブネット番号およびマスクの設定：例

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
exclude 1101:1::100 ::
!
!
end
```

IPv6 アドレスの範囲の指定

プール内の IPv6 アドレスの範囲を指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **address-range** *low_ip_address high_ip_address*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 addr_vrf`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。
ステップ 3	**address-range** *low_ip_address high_ip_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# address-range 1234::2 1234::3e81`	プール内の IPv6 アドレスの範囲を指定します。プール内では複数のアドレス範囲が許可されます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーショ

	コマンドまたはアクション	目的
		ンセッションを継続するには、**commit** コマンドを使用します。

IPv6 アドレスの範囲の指定：例

```
configure
pool vrf vrf1 ipv6 addr_vrf
address-range 1234::2 1234::3e81
!
!
end
```

使用率のしきい値の指定

プール IPv6 サブモードで特定の VRF に使用率のしきい値を指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ **1**	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。
ステップ 3	**prefix-length** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	クライアントに割り当てられたプレフィックスの長さを指定します。プレフィックス長の値の範囲は 1 ～ 128 です。
ステップ 4	**network** *subnet* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	サブネット内の一連のアドレスまたはプレフィックスを指定します。 （注） **prefix-length** コマンドは、**network** コマンドが使用される場合は、必ず設定する必要があります。
ステップ 5	**utilization-mark high** *value* **low** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	プール IPv6 サブモードで使用率のしきい値を指定します。 高い値および低い値は 0 ～ 100 のパーセンテージとして表されます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーショ

	コマンドまたはアクション	目的
		ン セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

使用率のしきい値の指定：例

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
!
!
end
```

プレフィックスの長さの指定

クライアントに割り当てられるプレフィックスの長さを指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **prefix-range** *low_ipv6_prefix high_ipv6_prefix*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 prefix_vrf`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。
ステップ 3	**prefix-length** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 64`	クライアントに割り当てられたプレフィックスの長さを指定します。 プレフィックス長の値の範囲は 1 ～ 128 です。
ステップ 4	**prefix-range** *low_ipv6_prefix high_ipv6_prefix* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-range 9fff:1:: 9fff:1:0:3e7f::`	プール IPv6 コンフィギュレーション モードで特定の VRF の IPv6 アドレス プレフィックスの範囲を指定します。 （注） **prefix-length** は **prefix-range** が設定される場合は、必ず設定する必要があります。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

クライアントに割り当てられるプレフィックスの長さの指定：例

```
configure
pool vrf vrf1 ipv6 prefix_vrf
prefix-length 64
prefix-range 9fff:1:: 9fff:1:0:3e7f::
!
!
end
```

サブネット内の一連のアドレスまたはプレフィックスの指定

プールIPv6コンフィギュレーションサブモードでサブネット内の一連のアドレスまたはプレフィックスを指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. **exclude** *low_ip_address high_ip_address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* 例： `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	特定の VRF の IPv6 プール名を作成し、プール IPv6 コンフィギュレーション サブモードを開始します。
ステップ 3	**prefix-length** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	クライアントに割り当てられたプレフィックスの長さを指定します。プレフィックス長の値の範囲は 1 ～ 128 です。
ステップ 4	**network** *subnet* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	サブネット内の一連のアドレスまたはプレフィックスを指定します。 （注） **prefix-length** コマンドは、**network** コマンドが使用される場合は、必ず設定する必要があります。
ステップ 5	**utilization-mark high** *value* **low** *value* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	プール IPv6 サブモードで使用率のしきい値を指定します。 高い値および低い値は 0 ～ 100 のパーセンテージとして表されます。
ステップ 6	**exclude** *low_ip_address high_ip_address* 例： `RP/0/RSP0/CPU0:router(config-pool-ipv6)# exclude 1101:1::100 ::`	DAPS がクライアントに割り当てることはできない IPv6 アドレスまたはプレフィックスの範囲を指定します。 高い値および低い値は 0 ～ 100 のパーセンテージとして表されます。 （注） プール内では複数の exclude コマンドが許可されます。1 つのアドレスを除外するには、<high_ip_address> を省略できます。
ステップ 7	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		`before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

サブネット内の一連のアドレスまたはプレフィックスの指定：例

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
exclude 1101:1::100 ::
!
!
end
```

PBR を使用した HTTP リダイレクト

HTTP リダイレクト（HTTPR）機能は、最初に指定された宛先以外の宛先に加入者トラフィックをリダイレクトするために使用されます。HTTPR 機能は、ルーティング プロトコルの代わりに、ポリシー設定に基づいてパケットの転送先を決定するポリシー ベース ルーティング（PBR）を使用して実装されます。 HTTPR 機能は、リダイレクト URL を含む HTTP リダイレクト応答を最初に要求を送信した HTTP クライアントに返信することによって実装されます。 その後、HTTP ク

ライアントは、リダイレクトされた URL に要求を送信します。 HTTPR は、IPv4 および IPv6 加入者の両方でサポートされます。

HTTPR 機能の最も一般的な用途は、最初のログイン用です。場合によっては、加入者を一意に識別し、許可することはできません。 これは、加入者がネットワークに接続するために、共有ネットワーク アクセス メディアを使用している場合に発生します。 このような場合、加入者は、ネットワークにアクセスできますが、「オープンガーデン」と呼ばれる機能に制限されます。 オープンガーデンは、加入者がネットワークへの物理アクセス権を持つ限りアクセスできるネットワーク リソースの集まりです。 加入者は、オープンガーデンの Web サイトにアクセスする前に、認証情報を入力する必要はありません。

加入者がオープンガーデン外のリソース（「ウォールドガーデン」と呼ばれます）にアクセスを試みると、Web のログイン ポータルにリダイレクトされます。 ウォールドガーデンは、加入者が最小の認証情報を入力してアクセスできる Web サイトまたはネットワークの集まりです。 Web のログイン ポータルでは、加入者はユーザ名とパスワードを使用してログインする必要があります。 その後、Web のログイン ポータルは、ユーザ クレデンシャルとともに BNG にアカウントログイン CoA を送信します。 これらのクレデンシャルの認証に成功すると、BNG はリダイレクトをディセーブルにし、直接のネットワーク アクセスに対して適切な加入者ポリシーを適用します。 HTTPR のその他の用途には、広告のための Web ポータルへの定期的なリダイレクション、課金サーバへのリダイレクションなどがあります。

PBR 機能は、独自の動的なテンプレートで設定されます。 動的なテンプレートに他の機能も含まれている場合、CoA を使用して、パケットをリダイレクトする PBR ポリシーを非アクティブ化する必要があります。

BNG は、リダイレクトまたはドロップされるパケット数を追跡する HTTP リダイレクト統計情報カウンタを維持します。 HTTP プロトコルは、いくつかのステータス コードを使用して HTTPR を実装します。 現在、リダイレクト コード 302（HTTP バージョン 1.0 の場合）および 307（HTTP バージョン 1.1 の場合）が BNG でサポートされています。

（注）

- HTTP リダイレクトは、HTTP パケットにのみ適用されます。 その結果、SMTP、FTP など他のサービスはこの機能に影響されません。 それにもかかわらず、これらの他のサービスがリダイレクト分類ルールの一部である場合、パケットはドロップされ、転送されません。
- HTTPS はサポートされていません。
- 宛先 URL ベースの分類はサポートされていません。

HTTPR の設定プロセスには、次の段階があります。

- リダイレクトされた権限およびオープンガーデン権限を定義するアクセス リストの作成。リダイレクションに対する HTTP の宛先の識別, （330 ページ）を参照してください。
- アクセス リストを使用してトラフィックをオープンガーデンへのアクセスがリダイレクトまたは許可されているものとして分類するクラスマップの作成。HTTP リダイレクションのクラス マップの設定, （335 ページ）を参照してください。

- クラスマップを使用して分類されたトラフィックで実行されるアクションを定義するポリシーマップの作成。HTTP リダイレクトのポリシーマップの設定，（337ページ）を参照してください。
- サービス ポリシーを適用するための動的なテンプレートの作成。HTTPR ポリシーを適用するための動的なテンプレートの設定，（340 ページ）を参照してください。

認証を実行する時間制限を指定する Web ログインを設定するには、Web ログインの設定，（342 ページ）を参照してください。

リダイレクションに対する HTTP の宛先の識別

リダイレクションを必要とするか、オープンガーデンの一部である HTTP の宛先を識別するアクセス リストを定義するには、次の作業を実行します。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**}**access-list** *redirect_acl_name*
3. 次のいずれかを実行します。
   - [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
4. 必要に応じてステップ 3 を繰り返し、シーケンス番号順にステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。
5. {**ipv4** | **ipv6**}**access-list** *open_garden_acl*
6. 次のいずれかを実行します。
   - [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
7. 必要に応じてステップ 6 を繰り返し、シーケンス番号順にステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

ステップ 2	{**ipv4** | **ipv6**}**access-list** *redirect_acl_name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-lists redirect_acl`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-lists redirect_acl`
		IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、名前付きアクセスリストを設定します。

ステップ 3	次のいずれかを実行します。
	• [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		IPv4 または IPv6 アクセス リストの redirect_acl で許可または拒否される 1 つまたは複数の条件を指定します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。
		
		または
		
		IPv6 アクセス リストの redirect_acl で許可または拒否される 1 つまたは複数の条件を指定します。
		• IPv6 オプション ヘッダーおよび任意の上位層プロトコル タイプ情報に基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**deny**（IPv6）コマンドおよび **permit**（IPv6）コマンドを参照してください。
		
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 4	必要に応じてステップ 3 を繰り返し、シーケンス番号順にステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。

	コマンドまたはアクション	目的

ステップ 5	{**ipv4** | **ipv6**}**access-list** *open_garden_acl*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-lists open_garden_acl`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-lists open_garden_acl`
		IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、オープンガーデンの名前付きアクセス リストを設定します。

ステップ 6	次のいずれかを実行します。
	• [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		IPv4 アクセス リストの open_garden_acl で許可または拒否される 1 つまたは複数の条件を指定します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。
		
		または
		
		IPv6 アクセス リストの open_garden_acl で許可または拒否される 1 つまたは複数の条件を指定します。
		• IPv6 オプション ヘッダーおよび任意の上位層プロトコル タイプ情報に基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**deny**（IPv6）コマンドおよび **permit**（IPv6）コマンドを参照してください。
		
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 7	必要に応じてステップ 6 を繰り返し、シーケンス番号順にステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。
ステップ 8	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

リダイレクションに対する HTTP の宛先の識別：例

```
configure
 ipv4 access-list <redirect-acl>
   10 permit tcp any any syn eq www
   20 permit tcp any any ack eq www
   30 permit tcp any any eq www
  ipv4 access-group <allow-acl>
   10 permit tcp any 10.1.1.0 0.0.0.255 eq www
   20 permit tcp any 20.1.1.0 0.0.0.255 eq www
   30 permit tcp any 30.1.1.0 0.0.0.255 eq www
   40 permit udp any any eq domain
 !
!
!
end

configure
 ipv6 access-list <redirect-acl>
   10 permit tcp any any syn eq www
   20 permit tcp any any ack eq www
   30 permit tcp any any eq www
  ipv6 access-group <allow-acl>
   10 permit tcp any 10.1.1.0 0.0.0.255 eq www
   20 permit tcp any 20.1.1.0 0.0.0.255 eq www
   30 permit tcp any 30.1.1.0 0.0.0.255 eq www
   40 permit udp any any eq domain
 !
```

```
!
!
end
```

HTTP リダイレクションのクラス マップの設定

HTTP リダイレクションのクラス マップを設定するには、次の作業を実行します。 以前に定義した ACL を使用します。

はじめる前に

リダイレクションに対する HTTP の宛先の識別, （330 ページ）で説明される設定手順は、HTTPR クラス マップの設定を実行する前に完了する必要があります。

手順の概要

1. **configure**
2. **class-map type traffic match-all** *open-garden-class_name*
3. **match** [**not**] **access-group**{**ipv4** | **ipv6**} *open_garden_acl*
4. **end-class-map**
5. **class-map type traffic match-all** *http_redirect-class_name*
6. **match** [**not**] **access-group** {**ipv4** | **ipv6**} *redirect_acl*
7. **end-class-map**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**class-map type traffic match-all** *open-garden-class_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type traffic match-all CL1`	トラフィック クラスと、パケットをオープン ガーデン クラスのクラスに一致させる関連ルールを定義します。

	コマンドまたはアクション	目的

ステップ 3	**match** [**not**] **access-group**{**ipv4** | **ipv6**} *open_garden_acl*
	
	例：
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv4 open_garden_acl`
	または
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv6 open_garden_acl`
		指定したアクセス コントロール リスト（ACL）番号をクラス マップの一致基準として識別します。
		（注） この手順で提供されるリダイレクト ACL 名は、前提条件で説明された設定手順で設定された名前です。

ステップ 4	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	クラスの一致基準の設定を終了し、クラス マップ コンフィギュレーション モードを終了します。
ステップ 5	**class-map type traffic match-all** *http_redirect-class_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type traffic match-all RCL1`	トラフィック クラスと、パケットをオープン ガーデン クラスのクラスに一致させる関連ルールを定義します。

ステップ 6	**match** [**not**] **access-group** {**ipv4** | **ipv6**} *redirect_acl*
	
	例：
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv4 redirect-acl`
	または
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv6 redirect-acl`
		指定したアクセス コントロール リスト（ACL）番号をクラス マップの一致基準として識別します。
		（注） この手順で提供されるリダイレクト ACL 名は、前提条件で説明された設定手順で設定された名前です。

ステップ 7	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	クラスの一致基準の設定を終了し、クラス マップ コンフィギュレーション モードを終了します。
ステップ 8	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` ◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コン

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	フィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

HTTP リダイレクションのクラス マップの設定：例

```
configure
class-map type traffic [match-any | match-all] <open-garden-class>
match [not] access-group ipv4 allow-acl
end-class-map

class-map type traffic [match-any | match-all] <http-redirect-class>
match [not] access-group ipv4 redirect-acl
end-class-map
  !
 !
!
end

configure
class-map type traffic [match-any | match-all] <open-garden-class>
match [not] access-group ipv6 allow-acl
end-class-map

class-map type traffic [match-any | match-all] <http-redirect-class>
match [not] access-group ipv6 redirect-acl
end-class-map
  !
 !
!
end
```

HTTP リダイレクトのポリシー マップの設定

HTTP リダイレクトのポリシー マップを設定するには、次の作業を実行します。

はじめる前に

リダイレクションに対する HTTP の宛先の識別, （330 ページ）およびHTTP リダイレクションのクラス マップの設定, （335 ページ）で説明される設定手順は、HTTPR のポリシーマップの設定を実行する前に完了する必要があります。

手順の概要

1. **configure**
2. **policy-map type pbr** *http-redirect_policy_name*
3. **class type traffic** *open_garden_class_name*
4. **transmit**
5. **class type traffic** *http_redirect-class_name*
6. **http-redirect** *redirect_url*
7. **class class-default**
8. **drop**
9. **end-policy-map**
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**policy-map type pbr** *http-redirect_policy_name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type pbr RPL1`	1 つまたは複数のインターフェイスに接続してサービス ポリシーを指定できるポリシーベース ルーティング タイプのポリシー マップを作成します。
ステップ 3	**class type traffic** *open_garden_class_name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic CL1`	ポリシーを作成または変更するクラスの名前を指定します。 （注） この手順で提供されるオープン ガーデンの ACL 名は、前提条件で説明された設定手順で設定された名前です。

	コマンドまたはアクション	目的
ステップ 4	**transmit** 例： `RP/0/RSP0/CPU0:router(config-pmap)# transmit`	元の宛先にパケットを転送します。
ステップ 5	**class type traffic** *http_redirect-class_name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic RCL1`	ポリシーを作成または変更するクラスの名前を指定します。 （注） この手順で提供されるオープン ガーデンのACL名は、前提条件で説明された設定手順で設定された名前です。
ステップ 6	**http-redirect** *redirect_url* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# http-redirect redirect_url`	HTTP要求がリダイレクトされる必要があるURLを指定します。
ステップ 7	**class class-default** 例： `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	ユーザ定義クラスでは使用できないデフォルトクラスを設定します。
ステップ 8	**drop** 例： `RP/0/RSP0/CPU0:router(config-pmap)# drop`	パケットをドロップします。
ステップ 9	**end-policy-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-policy-map`	ポリシー マップの設定を終了し、ポリシー マップコンフィギュレーションモードを終了します。

ステップ 10	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end**コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

HTTP リダイレクトのポリシー マップの設定：例

```
configure
policy-map type pbr <http-redirect-policy>
class type traffic <open-garden-class>
transmit
   !
class type traffic <http-redirect-class>
http-redirect <redirect-url>
!
class class-default
drop
  !
end-policy-map
  !
 !
!
end
```

HTTPR ポリシーを適用するための動的なテンプレートの設定

加入者セッションにHTTPRポリシーを適用するための動的なテンプレートを設定するには、次の作業を実行します。

はじめる前に

HTTP リダイレクトのポリシー マップの設定, （337 ページ）で説明される設定手順は、以前定義したポリシーマップを使用する動的なテンプレートを定義する前に完了する必要があります。

（注） 動的なテンプレートにポリシー ベース ルーティング ポリシーのみが含まれているため、Web ログインの後に容易に非アクティブにできることを確認します。

手順の概要

1. **configure**
2. **dynamic-template type ipsubscriber** *redirect_template_name*
3. **service-policy type pbr** *http-redirect-policy*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**dynamic-template type ipsubscriber** *redirect_template_name* 例： `RP/0/RSP0/CPU0:router(config)# dynamic-template type ipsubscriber RDL1`	「ipsubscriber」タイプの動的なテンプレートを作成します。
ステップ 3	**service-policy type pbr** *http-redirect-policy* 例： `RP/0/RSP0/CPU0:router(config-pmap)# service-policy type pbr RPL1`	以前の設定で作成されたポリシー マップ内の pbr タイプとしてサービス ポリシーを適用します。 （注） この手順で提供されるリダイレクト ポリシー名は、前提条件で説明された設定手順で設定された名前です。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ no と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーションファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

HTTPR ポリシーを適用するための動的なテンプレートの設定：例

```
configure
dynamic-template type ip <redirect-template>
service-policy type pbr <http-redirect-policy>
  !
  !
!
end
```

Web ログインの設定

Web ログイン設定するには、次の作業を実行します。 たとえば、タイマーは認証に許可される最大時間を定義します。

手順の概要

1. **configure**
2. **class-map type control subscriber match-all** *classmap_name*
3. **match timer** *name*
4. **match authen-status authenticated**
5. **policy-map type control subscriber** *policymap_name*
6. **event session-start match-all**
7. **class type control subscriber** *class_name* **do-until-failure**
8. *sequence_number* **activate dynamic-template** *dt_name*
9. *sequence_number* **activate dynamic-template** *dt_name*
10. *sequence_number* **set-timer** *timer_name value*
11. **event account-logon match-all**
12. **class type control subscriber** *class_name* **do-until-failure**
13. *sequence_number* **authenticate aaa list default**
14. *sequence_number* **deactivate dynamic-template***dt_name*
15. *sequence_number* **stop-timer** *timer_name*
16. **event time-expiry match-all**
17. **class type control subscriber** *class_name* **do-all**
18. *sequence_number* **disconnect**
19. 次のいずれかのコマンドを使用します。
    - **end**
    - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**class-map type control subscriber match-all** *classmap_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all IP_UNATH_COND`	match-all一致基準で加入者のコントロール クラスマップを設定します。

	コマンドまたはアクション	目的
ステップ 3	**match timer** *name* 例： `RP/0/RSP0/CPU0:router(config-cmap)# match timer AUTH_TIMER`	タイマーの詳細とともにクラスの一致基準を設定します。
ステップ 4	**match authen-status authenticated** 例： `RP/0/RSP0/CPU0:router(config-cmap)# match timer AUTH_TIMER`	認証ステータスの詳細とともにクラスの一致基準を設定します。
ステップ 5	**policy-map type control subscriber** *policymap_name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all RULE_IP_WEBSESSION`	加入者コントロール ポリシーマップを設定します。
ステップ 6	**event session-start match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	一致したクラスのすべてを実行するセッションの開始ポリシー イベントを設定します。
ステップ 7	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	加入者が一致するクラスを設定します。 一致があると、障害が見つかるまですべてのアクションを実行します。
ステップ 8	*sequence_number* **activate dynamic-template** *dt_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 activate dynamic-template DEFUALT_IP_SERVICE`	指定された動的なテンプレート名を使用して CLI でローカルに定義される動的なテンプレートをアクティブ化します。
ステップ 9	*sequence_number* **activate dynamic-template** *dt_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 activate dynamic-template HTTP_REDIRECT`	指定された動的なテンプレート名を使用して CLI でローカルに定義される動的なテンプレートをアクティブ化します。
ステップ 10	*sequence_number* **set-timer** *timer_name value* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 set-timer AUTH_TIMER 4567`	期限切れでルールを実行するようにタイマーを設定します。 分単位で指定されたタイマーの値の範囲は 0 ～ 4294967295 です。

	コマンドまたはアクション	目的
ステップ 11	**event account-logon match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	一致したクラスのすべてを実行するアカウント ログイン ポリシー イベントを設定します。
ステップ 12	**class type control subscriber** *class_name* **do-until-failure** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	加入者が一致するクラスを設定します。 一致があると、障害が見つかるまですべてのアクションを実行します。
ステップ 13	*sequence_number* **authenticate aaa list default** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 authenticate aaa list default`	デフォルトの AAA 方式リストを指定し、認証します。
ステップ 14	*sequence_number* **deactivate dynamic-template***dt_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 deactivate dynamic-template HTTP_REDIRECT`	期限切れになる前にタイマーをディセーブルにします。
ステップ 15	*sequence_number* **stop-timer** *timer_name* 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 20 stop-timer AUTH_TIMER`	期限切れになる前にタイマーをディセーブルにします。
ステップ 16	**event time-expiry match-all** 例： `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	一致したクラスのすべてを実行するタイマーの期限切れポリシー イベントを設定します。
ステップ 17	**class type control subscriber** *class_name* **do-all** 例： `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber IP_UNAUTH_COND do-all`	加入者が一致する必要があるクラスを設定します。 一致があると、すべてのアクションを実行します。
ステップ 18	*sequence_number* **disconnect** 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 disconnect`	セッションを切断します。
ステップ 19	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

Web ログインの設定：例

次の例では、クレデンシャルの認証 Web ポータルに HTTP リダイレクトされる IP セッションを示します。正常な認証では、タイマーの設定は解除されます。それ以外の場合は、タイマー ウィンドウが期限切れになったときに加入者が切断します。

```
class-map type control subscriber match-all IP_UNAUTH_COND
   match timer AUTH_TIMER
   match authen-status unauthenticated

policy-map type control subscriber RULE_IP_WEBSESSION
    event session-start match-all
        class type control subscriber class-default do-until-failure
            10 activate dynamic-template DEFAULT_IP_SERVICE
            20 activate dynamic-template HTTP_REDIRECT
            30 set-timer AUTH_TIMER 5

    event account-logon match-all
        class type control subscriber class-default do-until-failure
            10 authenticate aaa list default
            15 deactivate dynamic-template HTTP_REDIRECT
            20 stop-timer AUTH_TIMER
```

```
event timer-expiry match-all
    class type control subscriber IP_UNAUTH_COND do-all
        10 disconnect
```

その他の関連資料

ここでは、BNG の加入者機能の実装に関連する参考資料を示します。

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

付 録 A

BNG 機能の XML サポート

AAA、DHCP ポリシー プレーン、PPPoE、DAPS、および加入者データベースなどのほとんどの BNG 機能は、XML ベースのルータ コンフィギュレーションをサポートします。 Cisco XML API は、ルータの設定またはルータの設定、管理、および操作に関する情報の要求に使用できます。 Cisco XML API の使用の詳細については、http://www.cisco.com/en/US/products/ps9853/products_programming_reference_guides_list.htmlに記載されている『*Cisco IOS XR XML API Guide*』の最新リリースを参照してください。

Cisco XML API は、XML コマンドを使用してルータを設定します。 次の項に、BNG 機能でサポートされる XML コマンドを示します。

AAA XML サポート

XML のサポートは、アカウンティングと認証要求統計情報を取得する RADIUS で使用可能です。 AAA コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
radius-server dead-criteria time	`AAA.RADIUS. DeadCriteria.Time`
radius-server dead-criteria tries	`AAA.RADIUS. DeadCriteria.Tries`

CLI	XML
radius-server ipv4 dscp <value>	AAA.RADIUS. IPv4.DSCP
radius-server key {0 | 7 | LINE}	AAA.RADIUS.Key
radius-server retransmit <limit>	AAA.RADIUS.Retransmit
radius-server timeout <number>	AAA.RADIUS.Timeout
radius-server source-port extended	AAA.RADIUS.SourcePort.Extended
radius-server deadtime	AAA.RADIUS.DeadTime
radius-server load-balance method least-outstanding	AAA.RADIUS.LoadBalance.Method.LeastOutstanding
radius-server attribute list <attribute-name>	AAA.RADIUS.AttributeListTable.AttributeList.Enable
radius-server attribute list <attribute-name> attribute <radius-attributes>	AAA.RADIUS.AttributeListTable.AttributeList.Attribute
radius-server vsa attribute ignore unknown	AAA.RADIUS.VSA.Attribute.Ignore.Unknown
Radius-server host <> retransmit	AAA.RADIUS.HostTable.Host.Retransmit
Radius-server host <> timeout	AAA.RADIUS.HostTable.Host.Timeout
radius-server host <> key {0 | 7 | LINE}	AAA.RADIUS.HostTable.Host.Key
aaa server radius dynamic-author client <ip-address> vrf <vrf-name> server-key {0 | 7 | LINE}	AAA.RADIUS.DynamicAuthorization.ClientTable.Client.ServerKey

CLI	XML
aaa server radius dynamic-author ignore {server key | session key }	AAA.RADIUS.DynamicAuthorization.Ignore
aaa server radius dynamic-author port <port num>	AAA.RADIUS.DynamicAuthorization.Port
aaa accounting system default start-stop [broadcast] {group {radius | NAME1}} [group NAME2..] aaa accounting system rp-failover default start-stop [broadcast] {group {radius | NAME1}} [group NAME2..	AAA.AccountingTable.Accounting
aaa radius attribute nas-port-id format FORMAT_NAME	AAA.RADIUSAttribute.NASPortID.Format
aaa group server radius <group-name> { authorization } { reply | reject} <name>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Authorization.Reply
aaa group server radius <group-name> { authorization} { accept | request } <name>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Authorization.Request
aaa group server radius <group-name> { accounting } { accept | request} <name>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Accounting.Request
aaa group server radius	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Accounting.Reply

CLI	XML
<group-name> { accounting } { reply | reject} <name>	
aaa group server radius <group-name> load-balance method least-bounding	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.LoadBalance.Method.LeastBounding
aaa group server radius group1 source-interface	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.SourceInterface
aaa group server radius <radius-group> vrf <>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.VRF
aaa group server radius <radius-group> deadtime <>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.DeadTime
aaa group server radius <> server-private <host>	AAA.ServerGroups.RADIUSGroupTable.RADIUSGroup.PrivateServerTable.PrivateServer
show radius accounting	RADIUS.Accounting
show radius authentication	RADIUS.Authentication
show radius client	RADIUS.Client
show radius dynamic-author	RADIUS.DynamicAuthorization
show radius dead-criteria host <ip>	RADIUS.DeadCriteria.HostTable.Host
show radius server-groups	RADIUS.ServerGroups

DHCP XML サポート

XML のサポートは、クライアント バインディング、プロファイル情報、および DHCPv4 プロキシ統計情報を取得する DHCP で使用可能です。 これにより、管理クライアントは、Circuit-ID、Remote-ID、Mac-Address、ユーザ プロファイル情報、および DHCPv4 プロキシ統計情報に基づいてクライアント バインディングを実行できます。 DHCP コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
dhcp ipv4 profile <name> proxy relay information check	`DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.Check`

dhcp ipv4 profile <name>proxy relay information option[vpn | allow-untrusted | remote-id <name>]	`DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.AllowUntrusted`
	`DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.VPN`
	`DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.RemoteID`

dhcp ipv4 interface GigabitEthernet <interface-name> proxy profile <name>	`DHCPv4.InterfaceTable.Interface.Proxy.Profile`
dhcp ipv4 profile <name>proxy relay information policy [drop | keep | replace]	`DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.Policy`
dhcp ipv4 profile <name>proxy helper-address [vrf <name>] <server-ip-addr> [giaddr <ip-addr>]	`DHCPv4.ProfileTable.Profile.Proxy.VRFTable.VRF.HelperAddressTable.HelperAddress`
dhcp ipv4 profile <name> proxy broadcast-flag policy check	`DHCPv4.ProfileTable.Profile.Proxy.BroadcastFlag.Policy`

CLI	XML
dhcp ipv4 profile <name>proxy class <class-name> **helper-address [vrf <name>] <server-ip-addr> [giaddr <ip-addr>]** **match vrf <name> match option [124 | 125 | 60 | 77] hex <value> [mask <value>]**	DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.VRFTable.VRF. HelperAddressTable.HelperAddress DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.Match.VRF DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.Match.Option
dhcp ipv4 interface <interface> none	DHCPv4.InterfaceTable.Interface.None
dhcp ipv4 interface <interface> proxy [information option format-type circuit-id <cir-id>]	DHCPv4.InterfaceTable.Interface.Proxy.CircuitID
dhcp ipv4 vrf vrfname proxy profile <name>	DHCPv4.VRFTable.VRF
show dhcp ipv4 proxy binding circuit-id <cid> location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyCircuitIDFilter(Naming CircuitID)]
show dhcp ipv4 proxy binding remote-id <rid> location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyRemoteIDFilter(Naming RemoteID)]
show dhcp ipv4 proxy binding interface <ifSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyInterfaceFilter(Naming InterfaceName)]
show dhcp ipv4 proxy binding mac-address <addr> location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyMACAddressFilter(Naming MACAddress)]

CLI	XML
show dhcp ipv4 proxy binding location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyBriefFilter]
show dhcp ipv4 proxy binding detail location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable.Client
show dhcp ipv4 proxy binding summary location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.Binding.Summary
show dhcp ipv4 proxy binding vrf <vrfname>	DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4PProxyVRFFilter(Naming VRFName)]
show dhcp ipv4 proxy profile name <profile-name> location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.ProfileTable.Profile
show dhcp vrf <name> ipv4 proxy statistics location <locationSpecifier>	DHCPv4.NodeTable.Node.Proxy.VRFTable.VRF.Statistics
show dhcp ipv4 proxy statistics [location < loc >]	DHCPv4.NodeTable.Node.Proxy.Statistics

コントロール ポリシーの XML サポート

XML のサポートは、加入者管理、および加入者セッションの関連情報を取得するポリシープレーンで使用可能です。 コントロール ポリシー コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
interface <intf> service-policy type control subscriber <policy-name>	InterfaceConfigurationTable.InterfaceConfiguration.ControlSubscriber.ServicePolicy

CLI	XML
sh sub sess all loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable
sh sub sess all detail loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberDetailAllSessionFilter)
sh sub sess all summary loc <loc>	Subscriber.Session.NodeTable.Node.Summary
sh sub sess all username loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberAllUsenameFilter)
sh sub sess filter interface <intf-name> loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberInterfaceBriefFilter) {Naming InterfaceName}
sh sub sess filter interface <intf-name> detail loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberInterfaceDetailFilter) {Naming InterfaceName}
sh sub sess filter ipv4-address <IPv4-addr> loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address}
sh sub sess filter ipv4-address <IPv4-addr> detail loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address}
sh sub sess filter mac-address <mac-addr> loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberMACAddressBriefFilter) {Naming MACAddress}
sh sub sess filter mac-address <mac-addr> detail loc <loc>	Subscriber.Session.NodeTable.Node.SessionTable(SubscriberMACAddressDetailFilter) {Naming MACAddress}

CLI	XML
sh sub sess filter state <state> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberStateBriefFilter) {Naming State}`
sh sub sess filter state <state> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberStateDetailFilter) {Naming State}`
sh sub sess filter username <uname> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberUsernameBriefFilter) {Naming Username}`
sh sub sess filter username <uname> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberUsernameDetailFilter) {Naming Username}`
sh sub sess filter ipv4-address <IPv4 addr> vrf <vrf> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address}`
sh sub sess filter ipv4-address <IPv4-addr> vrf <vrf> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address}`
sh sub sess filter vrf <vrf-name> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address }`
sh sub sess filter vrf <vrf-name> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address }`
sh sub sess sub-label <0-ffffffff> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable.Session{Naming SessionID}`

CLI	XML
sh sub man stat AAA accounting loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.Accounting
sh sub man stat AAA accounting total loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAccounting
sh sub man stat AAA authentication loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.Authentication
sh sub man stat AAA authentication total loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAuthentication
sh sub man stat AAA authorization loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.Authorization
sh sub man stat AAA authorization total loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAuthorization
sh sub man stat AAA COA loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.ChangeOfAuthorization
sh sub man stat AAA COA total loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateChangeOfAuthorization
sh sub man stat AAA all loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA
sh sub man stat AAA all total loc <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AAA
sh sub man stats summary total <loc>	Subscriber.Manager.NodeTable.Node.Statistics.AggregateSummary

DAPS XML サポート

XML のサポートは、分散アドレス プール サービス（DAPS）のプール パラメータを取得する分散アドレスプールサービスで使用可能です。XMLのサポートによって、管理クライアントはVRFとプール名に基づいて無料で割り当ておよび除外されるアドレスの数を取得できます。 DAPS コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
pool vrf <vrf-name> ipv4 <poolname>pool ipv4 <poolname>	`PoolService.VRFTable.VRF.IPv4.Pool.Enable`
pool vrf <VRFName> ipv4 <PoolName> * address-range <RangeStart> <RangeEnd>pool ipv4 <PoolName> * address-range <RangeStart> <RangeEnd>	`PoolServce.VRFTable.VRF.IPv4.Pool.AddressRangeTable.AddressRange`
pool vrf <VRFName> ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>pool vrf <VRFName> ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>pool ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>	`PoolService.VRFTable.VRF.IPv4.Pool.ExcludeTable.Exclude`
Pool vrf <VRFName> ipv4 <PoolName> utilization-mark high <>pool ipv4 <PoolName> utilization-mark high <>	`PoolService.VRFTable.VRF.IPv4.Pool.UtilzationMark.High`
Pool vrf <VRFName> ipv4 <PoolName> utilization-mark low <>pool ipv4 <PoolName> utilization-mark low <>	`PoolService.VRFTable.VRF.IPv4.Pool.UtilzationMark.Low`
show pool vrf <vrf-name> ipv4	`PoolService.NodeTable.Node.VRFTable.VRF.IPv4`
show pool ipv4 name <poolname>	`PoolService.NodeTable.Node.PoolTable.Pool.IPv4.Detail`
show pool ipv4 name <poolname> verbose	`PoolService.NodeTable.Node.PoolTable.Pool.IPv4.Verbose`

CLI	XML
show pool ipv4show pool vrf all ipv4	`PoolService.NodeTable.Node.VRFTable`

PPPoE XML サポート

XML サポートは、PPP over Ethernet（PPPoE）セッションで使用可能です。 PPPoE 機能コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
pado delay {<delay>}	`set PadoDelay.Default {<delay>}`
pado delay circuit-id {<delay>}	`set PadoDelay.CircuitId {<delay>}`
pado delay remote-id {<delay>}	`set PadoDelay.RemoteId {<delay>}`
pado delay circuit-id string {<string>} {<delay>}	`set PadoDelay.CircuitIdString{<string>} {<delay>}`
pado delay circuit-id contains {<string>} {<delay>}	`set PadoDelay.CircuitIdSubString{<string>} {<delay>}`
pado delay remote-id string {<string>} {<delay>}	`set PadoDelay.RemoteIdString{<string>} {<delay>}`
pado delay remote-id contains {<string>} {<delay>}	`set PadoDelay.RemoteIdSubString{<string>} {<delay>}`
pado delay service-name string {<string>} {<delay>}	`set PadoDelay.ServiceNameString{<string>} {<delay>}`
pado delay service-name contains {<string>} {<delay>}	`set PadoDelay.ServiceNameSubString{<string>} {<delay>}`
pppoe session-id space flat	`set SessionIDSpaceFlat {TRUE}`
pppoe bba-group {<group-name>}	`PPPoECfg.BBAGroup {<group-name>}`
pppoe enable bba-group {<group-name>}	`set PPPoE.EnableBBAGroup {<group-name>}`
ac name {<name>}	`set Tags.ACName {<name>}`
service name {<name>}	`set Tags.ServiceName(<name>).ServiceNameConfigured`
service selection disable	`set Tags.ServiceSelectionDisable`
tag ppp-max-payload deny	`set Tags.PPPMaxPayloadDeny`
tag ppp-max-payload minimum {<min>} maximum {<max>}	`set Tags.PPPMaxPayload {<min>,<max>}`
mtu {<mtu>}	`set MTU {<mtu>}`
sessions max limit {limit} threshold {<threshold>}	`set Sesssions.MaxLimit {<limit>,<threshold>}`

CLI	XML
sessions access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.AccessInterfaceLimit {<count>,<threshold>}`
sessions mac limit {<count>} [threshold {<threshold>}]	`set Sessions.MacLimit {<count>,<threshold>}`
sessions mac-iwf limit {<count>} [threshold {<threshold>}]	`set Sessions.MacIWFLimit {<count>,<threshold>}`
sessions mac access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.MacAccessInterfaceLimit {<count>,<threshold>}`
sessions mac-iwf access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.MacIWFAccessInterfaceLimit {<count>,<threshold>}`
sessions circuit-id limit {<count>} [threshold {<threshold>}]	`set Sessions.CircuitIDLimit {<count>,<threshold>}`
sessions remote-id limit {<count>} [threshold {<threshold>}]	`set Sessions.RemoteIDLimit {<count>,<threshold>}`
sessions circuit-id-and-remote-id limit {<count>} [threshold {<threshold>}]	`set Sessions.CircuitIDAndRemoteIDLimit {<count>,<threshold>,<radius-override>}`
sessions inner-vlan limit {<count>} [threshold {<threshold>}]	`set Sessions.InnerVLANLimit {<count>,<threshold>}`
sessions mac throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions mac access-interface throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacAccessInterfaceThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions mac-iwf access-interface throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacIWFAccessInterfaceThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions circuit-id throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.CircuitIDThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions remote-id throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.RemoteIDThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions circuit-id-and-remote-id throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.CircuitIDAndRemoteIDThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions inner-vlan throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.InnerVLANThrottle {<request-count>,<request-period>,<blocking-period>}`
control-packets priority {<cos>}	`set ControlPackets.Priority {<cos>}`

CLI	XML
invalid-session-id drop	`set InvalidSessionID {DROP}`
invalid-session-id log	`set InvalidSessionID {LOG}`

加入者データベースの XML サポート

XML のサポートは、加入者のアソシエーション情報とセッション情報を取得する加入者データベースで使用可能です。XML のサポートにより、管理クライアントは、加入者セッションの状態、一意の加入者ラベルに基づく加入者のセッション情報、一意の加入者ラベルやインターフェイス名または動的なテンプレートの名前やタイプに基づく加入者のアソシエーション情報を取得できます。 加入者データベース コマンドの CLI および XML エントリ間のマッピングは、次の通りです。

CLI	XML
show subscriber database association br location <>	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label{Naming SubscriberLabel}`
show subscriber database association subscriber-label <> br location<>	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label{Naming SubscriberLabel}`
show subscriber database association location <>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseLabelDetailFilter)`
show subscriber database association interface-name <> br location<>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseInterfaceBriefFilter){Naming InterfaceName}`
show subscriber database association interface-name <> location<>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseInterfaceFilter){Naming InterfaceName}`
show subscriber database association type < ipsubscriber |ppp |service-profile |subscriber-service> br location <>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseTemplateTypeBriefFilter){Naming TemplateType}`
show subscriber database association type < ipsubscriber |ppp |service-profile |subscriber-service> location <>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseTemplateTypeFilter){Naming TemplateType}`

CLI	XML
show subscriber database session state <all| cfgapply |cfgdone |cfggen |cfgunapply |destroying |error |fatgen |init |sync>	Subscriber.Database.NodeTable.Node.Session(SubscriberDatabaseSessionStateFilter){Naming Session-State}
show subscriber database session subscriber-label <> location <>	Subscriber.Database.NodeTable.Node.Session.LabelTable.Label{Naming SubscriberLabel}
association subscriber-label <0x0-0xffffffff> brief location R/S/M	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label
association subscriber-label <0x0-0xffffffff> brief	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label
association subscriber-label <0x0-0xffffffff> location R/S/M	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label
association subscriber-label <0x0-0xffffffff>	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label
association interface-name <ifname> brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceBriefFilter(Naming InterfaceName]
association interface-name <ifname> brief	Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceBriefFilter(Naming InterfaceName]
association interface-name <ifname> location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceFilter(Naming InterfaceName]
association interface-name <ifname>	Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceFilter(Naming InterfaceName]
association type ppp brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ppp brief	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ppp location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ppp	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ipsubscriber brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]

CLI	XML
association type ipsubscriber brief	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ipsubscriber location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type ipsubscriber	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type subscriber-service brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type subscriber-service brief	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type subscriber-service location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type subscriber-service	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type service-profile brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type service-profile brief	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type service-profile location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type service-profile	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type user-profile brief location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type user-profile brief	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type user-profile location R/S/M	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association type user-profile	Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]
association brief location R/S/M	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label

CLI	XML
association brief	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association location R/S/M	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
session subscriber-label <0x0-0xffffffff> location R/S/M	`Subscriber.Database.NodeTable.Node.Session.LabelTable.Label`
session subscriber-label <0x0-0xffffffff>	`Subscriber.Database.NodeTable.Node.Session.LabelTable.Label`
session state init location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state init	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state destroying location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state destroying	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfggen location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfggen	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state fatgen location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state fatgen	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgapply location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgapply	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgdone location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgdone	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgunapply location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgunapply	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`

CLI	XML
session state cfgerror location R/S/M	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state cfgerror	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state error location R/S/M	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state error	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state sync location R/S/M	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state sync	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state all location R/S/M	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]
session state all	Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]

付 録 B

RADIUS 属性

リモート認証ダイヤルインユーザ サービス（RADIUS）属性は、RADIUS デーモンに保存されたユーザ プロファイル内の特定の認証、許可、アカウンティング（AAA）要素を定義するために使用されます。

この付録では、ブロードバンド ネットワーク ゲートウェイ（BNG）でサポートされる RADIUS 属性の次のタイプについて説明します。

RADIUS IETF 属性

IETF 属性と VSA の比較

RADIUS インターネット技術特別調査委員会（IETF）属性は、255 個の標準属性で構成されるオリジナルのセットで、クライアントとサーバ間での AAA 情報の伝達に使用されます。 IETF 属性は標準であるため、属性データは事前定義されてその内容も認識されています。このため、IETF 属性を介して AAA 情報を交換するすべてのクライアントとサーバは、属性の厳密な意味や各属性値の一般的な限界など、属性データに一致させる必要があります。

RADIUS ベンダー固有属性（VSA）は、1 つの IETF ベンダー固有属性（属性 26）から派生します。 属性 26 を使用すれば、ベンダーは、追加の 255 個の属性を自由に作成できます。 つまり、ベンダーは、どの IETF 属性のデータとも一致しない属性を作成して、属性 26 の背後にカプセル化することができます。そのため、新しく作成された属性は、属性26を受け入れているユーザに受け入れられます。

表 *6*：サポートされている *RADIUS IETF* 属性

名前	値	タイプ
Acct-Authentic	整数	45
Acct-Delay-Time	整数	41
Acct-Input-Giga-Words	整数	52
Acct-Input-Octets	整数	42
Acct-Input-Packets	整数	47
Acct-Interim-Interval	整数	85
Acct-Link-Count	整数	51
Acct-Output-Giga-Words	整数	53
Acct-Output-Octets	整数	43
Acct-Output-Packets	整数	48
Acct-Status-Type	整数	40
CHAP-Challenge	バイナリ	40
CHAP-Password	バイナリ	3
Dynamic-Author-Error-Cause	整数	101
Event-Timestamp	整数	55
Filter-Id	バイナリ	11
Framed-Protocol	整数	7
Framed-IP-Address	ipv4addr	8
Framed-Route	文字列	22
login-ip-addr-host	ipv4addr	14
Multilink-Session-ID	文字列	50
Nas-Identifier	文字列	32
NAS-IP-Address	ipv4addr	4
NAS-Port	整数	5
Reply-Message	バイナリ	18
Service-Type	整数	6
Tunnel-Assignment-Id	文字列	32

名前	値	タイプ
Tunnel-Packets-Lost	整数	86
X-Ascend-Client-Primary-DNS	ipv4addr	135
X-Ascend-Client-Secondary-DNS	ipv4addr	136
NAS-IPv6-Address	文字列	95
Delegated-IPv6-Prefix	バイナリ	123
Stateful-IPv6-Address-Pool	バイナリ	123
Framed-IPv6-Prefix	バイナリ	97
Framed-Interface-Id	バイナリ	96
Framed-IPv6-Pool	文字列	100
Framed-IPv6-Route	文字列	99
login-ip-addr-host	文字列	98

LAC の IETF タグ付き属性

L2TP アクセス コンセントレータ（LAC）の IETF タグ付き属性のサポートは、RADIUS サーバから LAC に送信される Access-Accept パケットで、同じトンネルを参照するトンネル属性をグループ化する手段を提供します。 Access-Accept パケットには、同じ RADIUS 属性でタグが異なる複数のインスタンスを含めることができます。 タグ付き属性のサポートは、指定のトンネルに属するすべての属性がそれぞれのタグ フィールドに同じ値を持ち、各セットに Tunnel-Preference 属性の適切な値のインスタンスが含まれるようにします。 これは、マルチベンダー ネットワーク環境で使用されるトンネル属性に準拠しているため、異なるベンダーで製造されたネットワーク アクセス サーバ（NAS）間の相互運用性の問題が解消されます。

トンネル プロトコル サポートの RADIUS 属性の詳細については、RFC 2868 を参照してください。

次の例で、IETF タグ付き属性の形式について説明します。

```
Tunnel-Type = :0:L2TP, Tunnel-Medium-Type = :0:IP, Tunnel-Server-Endpoint = :0:"1.1.1.1",
Tunnel-Assignment-Id = :0:"1", Tunnel-Preference = :0:1, Tunnel-Password = :0:"hello"
```

タグ値 0 は、上記の例で `:0:` の形式で使用されており、同じトンネルを参照する同じパケットでそれらの属性をグループ化します。 同様の例は、次のとおりです。

```
Tunnel-Type = :1:L2TP, Tunnel-Medium-Type = :1:IP, Tunnel-Server-Endpoint = :1:"2.2.2.2",
Tunnel-Assignment-Id = :1:"1", Tunnel-Preference = :1:1, Tunnel-Password = :1:"hello"

Tunnel-Type = :2:L2TP, Tunnel-Medium-Type = :2:IP, Tunnel-Server-Endpoint = :2:"3.3.3.3",
Tunnel-Assignment-Id = :2:"1", Tunnel-Preference = :2:2, Tunnel-Password = :2:"hello"

Tunnel-Type = :3:L2TP, Tunnel-Medium-Type = :3:IP, Tunnel-Server-Endpoint = :3:"4.4.4.4",
Tunnel-Assignment-Id = :3:"1", Tunnel-Preference = :3:2, Tunnel-Password = :3:"hello"
```

```
Tunnel-Type = :4:L2TP, Tunnel-Medium-Type = :4:IP, Tunnel-Server-Endpoint = :4:"5.5.5.5",
Tunnel-Assignment-Id = :4:"1", Tunnel-Preference = :4:3, Tunnel-Password = :4:"hello"

Tunnel-Type = :5:L2TP, Tunnel-Medium-Type = :5:IP, Tunnel-Server-Endpoint = :5:"6.6.6.6",
Tunnel-Assignment-Id = :5:"1", Tunnel-Preference = :5:3, Tunnel-Password = :5:"hello"
```

表 *7*：サポートされる *IETF* タグ付き属性

IETF タグ付き属性の名前	値	タイプ
Tunnel-Type	整数	64
Tunnel-Medium-Type	整数	65
Tunnel-Client-Endpoint	文字列	66
Tunnel-Server-Endpoint	文字列	67
Tunnel-Password	文字列	69
Tunnel-Assignment-ID	文字列	82
Tunnel-Preference	整数	83
Tunnel-Client-Auth-ID	文字列	90
Tunnel-Server-Auth-ID	文字列	91

RADIUS ベンダー固有属性

インターネット技術特別調査委員会（IETF）ドラフト標準には、ネットワーク アクセス サーバと RADIUS サーバの間でベンダー固有属性（属性 26）を使用してベンダー固有の情報を伝達する方法が規定されています。 属性 26 はベンダー固有属性をカプセル化します。このため、ベンダーは一般的な用途に適さない独自の拡張属性をサポートできます。

シスコの RADIUS 実装は、この仕様で推奨される形式を使用して、1 つのベンダー固有オプションをサポートしています。 シスコのベンダー ID は 9 で、サポートされるオプションはベンダータイプ 1、名前は「cisco-av-pair」です。値は次の形式の文字列になります。

```
protocol : attribute sep value *
```

「Protocol」は、特定の許可タイプを表すシスコの「protocol」属性です。使用可能なプロトコルには、IP、IPX、VPDN、VOIP、SHELL、RSVP、SIP、AIRNET、OUTBOUND があります。「attribute」および「value」は、シスコの TACACS+ 仕様で定義されている適切な属性値（AV）ペアです。「sep」は、必須の属性の場合は「=」、任意指定の属性の場合は「*」になります。 これにより、TACACS+許可で使用できるすべての機能を RADIUS にも使用できるようになります。

たとえば、次の AV ペアにより、IP を許可している間（PPP の IPCP アドレス割り当てを行っている間）、シスコの「指定された複数の IP アドレスプール」をアクティブにすることができます。

```
cisco-avpair= "ip:addr-pool=first"
```

「*」を挿入すると、AV ペア「ip:addr-pool=first」はオプションになります。 AV ペアはオプションにできることに注意してください。

IETF 属性 26（ベンダー固有）は、ベンダー固有属性をカプセル化します。このため、ベンダーは一般的な用途に適さない独自の拡張属性をサポートできます。

```
cisco-avpair= "ip:addr-pool*first"
```

次に、ネットワーク アクセス サーバからユーザがログインしたときに、すぐに EXEC コマンドを実行する方法の例を示します。

```
cisco-avpair= "shell:priv-lvl=15"
```

属性 26 には、次の 3 つの要素が含まれています。

- タイプ
- 長さ
- ストリング（またはデータ）
  - Vendor-Id
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

（注） VSA の形式はベンダーが指定します。 Attribute-Specific フィールド（Vendor-Data とも呼ばれる）は、ベンダーによるその属性の定義によって異なります。

表 *8*：サポートされるシスコのベンダー固有 *RADIUS* 属性

名前	値	タイプ
access-loop-encapsulation	バイナリ	1
accounting-list	文字列	1
acct-policy-in	文字列	1
acct-policy-map	文字列	1
acct-policy-out	文字列	1
actual-data-rate-downstream	整数	1
actual-data-rate-upstream	整数	1
actual-interleaving-delay-downstream	整数	1
actual-interleaving-delay-upstream	整数	1
attainable-data-rate-downstream	整数	1

名前	値	タイプ
attainable-data-rate-upstream	整数	1
circuit-id-tag	文字列	1
cisco-nas-port	文字列	2
client-mac-address	文字列	1
command	文字列	1
connect-progress	文字列	1
connect-rx-speed	整数	1
connect-tx-speed	整数	1
dhcp-client-id	文字列	1
dhcp-vendor-class	文字列	1
disc-cause-ext	文字列	1
Disconnect-Cause	文字列	1
if-handle	整数	1
inacl	文字列	1
interworking-functionality-tag	ブール	1
ip-addresses	文字列	1
ip-unnumbered	文字列	1
ipv4-unnumbered	文字列	1
login-ip-host	文字列	1
maximum-interleaving-delay-downstream	整数	1
maximum-interleaving-delay-upstream	整数	1
maximum-data-rate-downstream	整数	1
maximum-data-rate-upstream	整数	1
minimum-data-rate-downstream	整数	1
minimum-data-rate-downstream-low-power	整数	1
minimum-data-rate-upstream	整数	1
minimum-data-rate-upstream-low-power	整数	1
parent-if-handle	整数	1

名前	値	タイプ
pppoe_session_id	整数	1
qos-policy-in	文字列	1
qos-policy-out	文字列	1
redirect-vrf	文字列	1
remote-id-tag	文字列	1
service-acct-list	文字列	1
service-name	文字列	1
sub-qos-policy-in	文字列	1
sub-qos-policy-out	文字列	1
traffic-class	文字列	1
tunnel-tos-reflect	文字列	1
tunnel-tos-setting	整数	1
vpn-id	文字列	1
vpn-vrf	文字列	1
vrf-id	整数	1
ipv6-enable	整数	1
ipv6-mtu	整数	1
ipv6-strict-rpf	整数	1
ipv6-unreachable	整数	1
acct-input-gigawords-ipv6	整数	1
acct-input-octets-ipv6	整数	1
acct-input-packets-ipv6	整数	1
acct-output-gigawords-ipv6	整数	1
acct-output-octets-ipv6	整数	1
acct-output-packets-ipv6	整数	1
delegated-ipv6-pool	文字列	1
ipv6-dns-servers-addr	文字列	1
dhcpv6-class	文字列	1

名前	値	タイプ
ipv6_inacl	文字列	1
ipv6_outacl	文字列	1
addrv6	文字列	1
acct-input-gigawords-ipv4	整数	1
acct-input-octets-ipv4	整数	1
acct-input-packets-ipv4	整数	1
acct-output-gigawords-ipv4	整数	1
acct-output-octets-ipv4	整数	1
acct-output-packets-ipv4	整数	1

RADIUS ADSL 属性

表 *9*：サポートされる *RADIUS ADSL* 属性

名前	値	タイプ
Access-Loop-Encapsulation	バイナリ	144
Actual-Interleaving-Delay-Downstream	整数	142
Actual-Interleaving-Delay-Upstream	整数	140
Actual-Data-Rate-Downstream	整数	130
Actual-Data-Rate-Upstream	整数	129
Attainable-Data-Rate-Downstream	整数	134
Attainable-Data-Rate-Upstream	整数	133
Agent-Circuit-Id	文字列	1
IWF-Session	ブール ソーシャル	254
Maximum-Interleaving-Delay-Downstream	整数	141
Maximum-Interleaving-Delay-Upstream	整数	139
Maximum-Data-Rate-Downstream	整数	136
Maximum-Data-Rate-Upstream	整数	135
Minimum-Data-Rate-Downstream	整数	132

名前	値	タイプ
Minimum-Data-Rate-Downstream-Low-Power	整数	138
Minimum-Data-Rate-Upstream	整数	131
Minimum-Data-Rate-Upstream-Low-Power	整数	137
Agent-Remote-Id	文字列	2

RADIUS ASCEND 属性

表 *10*：サポートされる *RADIUS ASCEND* 属性

名前	値	タイプ
Ascend-Client-Primary-DNS	ipv4addr	135
Ascend-Client-Secondary-DNS	ipv4addr	136
Ascend-Connection-Progress	整数	196
Ascend-Disconnect-Cause	整数	195
Ascend-Multilink-Session-ID	整数	187
Ascend-Num-In-Multilink	整数	188

Microsoft RADIUS 属性

表 *11*：サポートされる *Microsoft RADIUS* 属性

名前	値	タイプ
MS-1st-NBNS-Server	ipv4addr	30
MS-2nd-NBNS-Server	ipv4addr	31
MS-CHAP-ERROR	バイナリ	2
MS-Primary-DNS	ipv4addr	28
MS-Secondary-DNS	ipv4addr	29

RADIUS Disconnect-Cause 属性

Disconnect-cause 属性値は、接続がオフラインにされた理由を指定します。 属性値は、Accounting 要求パケットで送信されます。 セッションの認証が失敗しても、これらの値は、セッションの終了時に送信されます。 セッションが認証されないと、属性が開始レコードを生成せずに終了レコードを発生させる可能性があります。

Disconnect-Cause（195）属性の原因コード、値、および説明を示します。

（注） Disconnect-Cause は、RADIUS AVPairs で使用されるごとに 1000 ずつ増分されます。たとえば、disc-cause 4 は 1004 になります。

表 *12*：サポートされる *Disconnect-Cause* 属性

原因コード	値	説明
0	No-Reason	接続解除の理由は提供されない。
1	No-Disconnect	イベントは接続解除されていない。
2	Unknown	理由は不明。
3	Call-Disconnect	コールが接続解除された。
4	CLID-Authentication-Failure	calling-party 数の認証の失敗。
9	No-Modem-Available	コールへの接続にモデムが使用できない。
10	No-Carrier	キャリアが検出されない。 （注） 最初のモデム接続中に接続解除があると、コード 10、11、および 12 が送信される場合があります。
11	Lost-Carrier	キャリアの喪失。
12	No-Detected-Result-Codes	モデム結果コード検出の失敗。

原因コード	値	説明
20	User-Ends-Session	ユーザがセッションを終了した。 （注） コード 20、22、23、24、25、26、27、および 28 は、EXEC セッションに適用されます。
21	Idle-Timeout	ユーザ入力待機中のタイムアウト。 （注） コード 21、100、101、102、および 120 は、すべてのセッション タイプに適用されます。
22	Exit-Telnet-Session	既存の Telnet セッションによる接続解除。
23	No-Remote-IP-Addr	SLIP/PPP への切り替え不能。リモート エンドに IP アドレスがない。
24	Exit-Raw-TCP	既存の raw TCP による接続解除。
25	Password-Fail	間違ったパスワード。
26	Raw-TCP-Disabled	Raw TCP がディセーブルにされた。
27	Control-C-Detected	Control-C が検出された。
28	EXEC-Process-Destroyed	EXEC プロセスが破棄された。
29	Close-Virtual-Connection	ユーザが仮想接続を終了した。
30	End-Virtual-Connection	仮想接続が終了した。
31	Exit-Rlogin	ユーザが Rlogin を終了した。
32	Invalid-Rlogin-Option	無効な Rlogin オプションが選択された。
33	Insufficient-Resources	不十分なリソース。

原因コード	値	説明
40	Timeout-PPP-LCP	PPP LCP ネゴシエーションがタイムアウトした。 （注） コード 40 ～ 49 が PPP セッションに適用されます。
41	Failed-PPP-LCP-Negotiation	PPP LCP ネゴシエーションが失敗した。
42	Failed-PPP-PAP-Auth-Fail	PPP PAP 認証が失敗した。
43	Failed-PPP-CHAP-Auth	PPP CHAP 認証が失敗した。
44	Failed-PPP-Remote-Auth	PPP リモート認証が失敗した。
45	PPP-Remote-Terminate	PPP がリモート エンドから Terminate Request を受信した。
46	PPP-Closed-Event	上位層がセッションの終了を要求した。
47	NCP-Closed-PPP	開いている NCP がなかったため、PPP セッションが終了した。
48	MP-Error-PPP	MP エラーのため、PPP セッションが終了した。
49	PPP-Maximum-Channels	最大チャネルに達したため、PPP セッションが終了した。
50	Tables-Full	ターミナル サーバ テーブルがいっぱいになったため、接続解除された。
51	Resources-Full	内部リソースがいっぱいになったため、接続解除された。
52	Invalid-IP-Address	Telnet ホストに対する IP アドレスが有効でない。
53	Bad-Hostname	ホスト名が検証されていない。
54	Bad-Port	ポート番号が無効または欠落している。

原因コード	値	説明
60	Reset-TCP	TCP 接続がリセットされた。 （注） コード 60 ～ 67 は Telnet または raw TCP セッションに適用されます。
61	TCP-Connection-Refused	TCP接続がホストによって拒否された。
62	Timeout-TCP	TCP接続がタイムアウトした。
63	Foreign-Host-Close-TCP	TCP 接続が終了した。
64	TCP-Network-Unreachable	TCPネットワークに到達できない。
65	TCP-Host-Unreachable	TCP ホストに到達できない。
66	TCP-Network-Admin Unreachable	管理上の理由により、TCPネットワークに到達できない。
67	TCP-Port-Unreachable	TCP ポートに到達できない。
100	Session-Timeout	セッションがタイムアウトした。
101	Session-Failed-Security	セキュリティ上の理由から、セッションが失敗した。
102	Session-End-Callback	コールバックにより、セッションが終了した。
120	Invalid-Protocol	検出されたプロトコルがディセーブルにされていたため、コールが拒否された。
150	RADIUS-Disconnect	RADIUS要求による接続解除。
151	Local-Admin-Disconnect	管理上の接続解除。
152	SNMP-Disconnect	SNMP 要求による接続解除。
160	V110-Retries	許可された V.110 リトライを超過した。
170	PPP-Authentication-Timeout	PPP認証がタイムアウトした。
180	Local-Hangup	ローカルのハングアップによって接続解除された。

原因コード	値	説明
185	Remote-Hangup	リモート エンドのハングアップよって接続解除された。
190	T1-Quiesced	T1 回線が休止状態のため接続解除された。
195	Call-Duration	コールの最大継続時間を超過したため、接続解除された。
600	VPN-User-Disconnect	クライアントによってコールが接続解除された（PPP経由）。LNS がクライアントから PPP terminate request を受信するとコードが送信されます。
601	VPN-Carrier-Loss	キャリアの喪失。 これは回線が物理的に普通になった結果である場合があります。 クライアントがダイヤラを使用してダイヤルアウトできない場合、コードが送信されます。
602	VPN-No-Resources	コールの処理に使用できるリソースがない。 クライアントがメモリを割り当てることができない場合、コードが送信されます（メモリの不足）。
603	VPN-Bad-Control-Packet	L2TP または L2F 制御パケットが間違っている。 このコードは、必須の属性値ペア（AVP）が欠落しているなど、ピアから受信した制御パケットが無効な場合に送信されます。 L2TP を使用すると、コードは6回の再送信後に送信されます。L2F を使用すると、再送信の回数はユーザ設定が可能です。 （注） トンネルにアクティブなセッションがある場合は、VPN-Tunnel-Shut が送信されます。

原因コード	値	説明
604	VPN-Admin-Disconnect	管理上の接続解除。 これは、VPN ソフト シャットダウンの結果である場合があります。これは、クライアントが最大セッション制限に達するか、最大ホップカウントを超過した場合に発生します。 トンネルが、clear vpdn tunnel コマンドの発行によってダウンした場合に、コードが送信されます。
605	VPN-Tunnel-Shut	トンネルのティアダウン、またはトンネルのセットアップが失敗した。 トンネルにアクティブなセッションがあり、トンネルがダウンした場合にコードが送信されます。 （注） このコードはトンネルの認証が失敗した場合は、送信されません。
606	VPN-Local-Disconnect	LNS PPP モジュールによって、コールが接続解除された。 LNS がクライアントに PPP terminate request を送信するとコードが送信されます。 これは通常の PPP 接続解除が LNS によって開始されたことを示します。
607	VPN-Session-Limit	VPN ソフト シャットダウンがイネーブルになった。 前述したソフト シャットダウンの制約事項のいずれかによってコールが拒否されると、コードが送信されます。
608	VPN-Call-Redirect	VPN コール リダイレクトがイネーブルになった。

付 録 C

アクション ハンドラ

アクション ハンドラは、特定のイベントに応じて特定の作業を実行します。 次のアクション ハンドラは、現在 BNG でサポートされています。

許可のアクション ハンドラ

許可のアクション ハンドラは、外部の AAA サーバから特定の加入者 ID の許可データを取得します。 許可のアクション ハンドラは、非同期関数です。 加入者属性データベース（SADB）、および CLI で指定された ID タイプに基づくユーザ クレデンシャル データから ID 情報を収集します。 この情報は、方式リストの名前とともに、AAA 許可コーディネータに送信されます。 AAA の処理が完了すると、制御はイベント処理を完了するためにポリシー ルール エンジン（PRE）アクション ハンドラに戻されます。 設定例は次のとおりです。

```
1 authorize aaa list <list-name> [identifier <identifier-type> | format <format_name>
password ['use-from-line'| <user-cfg-password>
```

（注） パスワードは、ユーザが回線からの使用を選択するか、許可に使用する特定の値を提供するかどうかに関係なく、必須です。

認証のアクション ハンドラ

認証のアクション ハンドラは、プロトコル タイプ、サービス タイプ、認証タイプ、ユーザ名、CHAP 属性、およびユーザ パスワードなどの情報を収集して、AAA 方式リストの名前とともに AAA のコーディネータに渡します。 認証のアクション ハンドラは、非同期関数です。 AAA の

処理が完了すると、制御はイベント処理を完了するために PRE アクション ハンドラに戻されます。 設定例は次のとおりです。

```
1 authenticate aaa list <list-name>
```

切断のアクション ハンドラ

切断のアクション ハンドラは、加入者を切断するためにコールされます。 加入者の切断に関して、PRE は、加入者の切断に関するすべてのクライアントを知らせるように、ポリシー プレーン セッション マネージャ（PPSM）に通知します。 PPSM は、切断を完了するために PRE に報告します。 PRE は、加入者を切断状態にします。 PRE は、ポリシー実行履歴と加入者のラベルを含む制御ブロックを保存するレコード履歴データも削除します。 PREの処理が実行されると、制御は処理のために PPSM に戻されます。

アクティブ化のアクション ハンドラ

アクティブ化のアクション ハンドラは、加入者設定のローカルな動的なテンプレートまたはリモートの AAA サービスをイネーブルにします。 このアクションの結果は、即時または非同期です。 PRE は、AAA 方式リスト名、テンプレート タイプ、およびテンプレート名などの情報を収集し、SVM に送信して処理します。 SVM はテンプレートの処理が終了した後に制御を戻し、PREは停止していた場所からアクション リストの処理を再開します。 設定例は次のとおりです。

```
1 activate dynamic-template <template-name> [aaa list <list-name>]
```

非アクティブ化のアクション ハンドラ

非アクティブ化のアクション ハンドラは、加入者設定からローカルの動的なテンプレートまたはリモートの AAA サービスをディセーブルにします。 このアクションの結果は、非同期です。 PRE は、AAA リスト、テンプレート タイプ、およびテンプレート名などの情報を収集して、SVM に送信します。 サービスを適用しないように要求します。 AAA リストは、SVM で使用されるキーを取得するために使用されます。 SVM はテンプレートの処理が終了した後に制御を戻し、PRE は停止していた場所からアクション リストの処理を再開します。 設定例は次のとおりです。

```
1 deactivate dynamic-template <template-name> [aaa list <list-name>]
```

タイマー設定とタイマー停止のアクション ハンドラ

タイマー設定のアクション ハンドラは、アクティブな名前付きタイマーを加入者セッションで定義された期間に設定します。 タイマー停止は、加入者セッションのアクティブな名前付きタイマーを停止します。 タイマー設定のアクション ハンドラをイネーブルにすると、サービス プロバイダーは加入者でトリガーされる設定時間のポリシー期限切れイベントを1つまたは複数持つことができます。 これは、その加入者のライフ サイクルにより優れた加入者管理を提供します。 これらのアクション ハンドラは、加入者ステートの状態のスケジュール化された確認（加入者が認証か非認証かを確認する）や加入者ポリシーの定期的な変更（毎日または時間ごとに再承認を強制するなど）といった機能を提供します。

（注） タイマー値が 0 のアクションは、アクションをすぐにトリガーします。

アクティブなタイマーを停止する方法は、次の 2 通りあります。

- タイマーが期限切れになるようにします。
- タイマー停止のアクション コマンドを使用して、実行中のアクティブなタイマーを停止します。

索 引

A

B

C

D

F

H

I

L

M

N

P

Q

R

S

T

U

V

あ

い

お

か

き

く

け

こ

さ

し

す

せ

と

な

に

は

ふ

へ

ほ

ま

ゆ

り