



Easy Virtual Network コンフィギュレーション ガイド

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/>

【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。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

シスコが導入する TCP ヘッダー圧縮は、カリフォルニア大学バークレー校（UCB）により、UNIX オペレーティングシステムの UCB パブリック ドメイン バージョンの一部として開発されたプログラムを適応したものです。All rights reserved.Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco および Cisco ロゴは、シスコまたはその関連会社の米国およびその他の国における商標または登録商標です。シスコの商標の一覧は、<http://www.cisco.com/go/trademarks> でご確認いただけます。掲載されている第三者の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語の使用はシスコと他社との間のパートナーシップ関係を意味するものではありません。(1110R)

© 2017 Cisco Systems, Inc. All rights reserved.



目次

最初にお読みください 1

Easy Virtual Network の概要 3

機能情報の確認 3

EVN の設定に関する前提条件 4

EVN の制約事項 4

EVN の概要 5

EVN の利点 5

仮想ネットワーク タグによるパスの分離の提供 6

仮想ネットワーク タグ 8

vnet Global 9

エッジ インターフェイスと EVN トランク インターフェイス 9

表示出力でのトランク インターフェイスの識別 11

トランク インターフェイスの単一の IP アドレス 11

定義されている VRF とトランク インターフェイス上で実行する VRF 間の関係 12

VRF 認識 12

EVN でサポートされるルーティングプロトコル 13

仮想ネットワークのパケットフロー 13

EVN トランク インターフェイスのコマンド継承 15

コマンド継承仮想ネットワーク インターフェイス モードの上書き 16

例：コマンド継承の上書き 16

例：vnet Global のみへの属性の有効化 16

上書きの削除と EVN トランクから継承された値の復元 17

コンフィギュレーションファイルにコマンドの no 形式が表示されるかどうかの判定 18

EXEC コマンドルーティング コンテキスト 18

EVN の VRF-Lite との互換性 19

マルチアドレス ファミリ VRF 構造 20

EVN での QoS 機能	20
インターフェイス上の仮想ネットワークにより値が継承または上書きできるコマンド	20
その他の参考資料	25
Easy Virtual Network の概要の機能情報	26
Easy Virtual Network の設定	27
機能情報の確認	27
EVN の設定に関する前提条件	28
EVN の設定方法	28
Easy Virtual Network トランク インターフェイスの設定	28
トランク インターフェイス上における VRF サブセットの有効化	34
EVN エッジ インターフェイスの設定	36
次の作業	37
EVN 設定の確認	38
EVN の設定例	39
例：network コマンドでの OSPF を使用した仮想ネットワーク	39
例：ip ospf vnet area コマンドでの OSPF を使用した仮想ネットワーク	40
例：EIGRP 環境でのコマンド継承と仮想ネットワーク インターフェイス モードの上書き	40
例：マルチキャスト環境でのコマンド継承と仮想ネットワーク インターフェイス モードの上書き	43
例：IP マルチキャストを使用した EVN	44
その他の参考資料	45
Easy Virtual Network の設定の機能情報	46
Easy Virtual Network の管理とトラブルシューティング	49
機能情報の確認	49
EVN の管理とトラブルシューティングの前提条件	50
EVN の管理とトラブルシューティングの概要	50
EXEC モードのルーティング コンテキストにより VRF の指定の繰り返しを減らす	50
traceroute コマンドの出力は VRF 名と VRF タグを示す	51
VRF 単位のデバッグ出力のフィルタ	51

CISCO-VRF-MIB	51
EVN の管理とトラブルシューティングの方法	51
特定の VRF への EXEC モードのルーティング コンテキストの設定	51
VRF のデバッグ出力のイネーブル化	53
仮想ネットワークの SNMP v2c コンテキストの設定	54
仮想ネットワークの SNMP v3 コンテキストの設定	55
その他の参考資料	56
EVN の管理とトラブルシューティングの機能情報	57
Easy Virtual Network 共有サービスの設定	59
機能情報の確認	59
仮想 IP ネットワーク共有サービスの前提条件	59
仮想 IP ネットワーク共有サービスの制約事項	60
Easy Virtual Network 共有サービスの概要	60
Easy Virtual Network の共有サービス	60
VRF-Lite よりも簡単な Easy Virtual Network の共有サービス	60
Easy Virtual Network のルート レプリケーション プロセス	61
ルート レプリケーションを実装する場所	62
Easy Virtual Network のルート レプリケーションの動作	62
Easy Virtual Network でのルート レプリケーション後のルート プリファレンス ルール	63
Easy Virtual Network を使用してサービスを共有する方法	63
Easy Virtual Network でサービスを共有するためのルート レプリケーションの設定	63
例	70
次の作業	71
Easy Virtual Network の共有サービスへの再配布の設定	71
Easy Virtual Network 共有サービスの設定例	74
例：マルチキャスト環境での Easy Virtual Network ルート レプリケーションとルート再配布	74
その他の参考資料	79
Easy Virtual Network 共有サービスの機能情報	80



第 1 章

最初にお読みください

Cisco IOS XE 16 に関する重要な情報

現行の Cisco IOS XE リリース 3.7.0E (Catalyst スイッチ用) および Cisco IOS XE リリース 3.17S (アクセスおよびエッジルーティング用) の 2 つのリリースは、1 つのバージョンの統合されたリリース (Cisco IOS XE 16) へと発展しています。これにより、スイッチングおよびルーティングポートフォリオの幅広い範囲のアクセスおよびエッジ製品に 1 つのリリースで対応できます。



(注)

技術設定ガイドの機能情報の表には、機能が導入された時期が示されています。その他のプラットフォームでその機能がサポートされた時期については示されていない場合があります。特定の機能がご使用のプラットフォームでサポートされているかどうかを特定するには、製品のランディング ページに示されている技術設定ガイドを参照してください。技術設定ガイドが製品のランディング ページに表示されている場合は、その機能がプラットフォームでサポートされていることを示します。



第 2 章

Easy Virtual Network の概要

Easy Virtual Network (EVN) は、複数のレイヤ 3 ネットワークのエンドツーエンドの仮想化を実現する IP ベースの仮想化テクノロジーです。単一の IP インフラストラクチャを使用して、トラフィック パスが相互に独立した状態で、個別の仮想ネットワークを提供できます。

EVN は、VRF-Lite と呼ばれる既存の IP ベースの仮想化メカニズムに基づいて構築されています。EVN はパス分離の拡張機能、簡単な設定と管理、改善された共有サービス サポートを提供します。EVN は VRF-Lite と後方互換性があり、VRF-Lite から EVN へのシームレスなネットワークの移行が可能です。

EVN は、IPv4、スタティック ルート、Open Shortest Path First version 2 (OSPFv2)、およびユニキャスト ルーティングでの Enhanced Interior Gateway Routing Protocol (EIGRP) と IPv4 マルチキャスト ルーティングでの PIM (Protocol Independent Multicast) および Multicast Source Discovery Protocol (MSDP) をサポートします。EVN はまた、Cisco Express Forwarding (CEF) と Simple Network Management Protocol (SNMP) もサポートします。

- [機能情報の確認, 3 ページ](#)
- [EVN の設定に関する前提条件, 4 ページ](#)
- [EVN の制約事項, 4 ページ](#)
- [EVN の概要, 5 ページ](#)
- [その他の参考資料, 25 ページ](#)
- [Easy Virtual Network の概要の機能情報, 26 ページ](#)

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、[Bug Search Tool](#) およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

EVN の設定に関する前提条件

- ネットワークへの EVN の実装には、複数の論理ネットワークまたは L3VPN に仮想化する、単一の IP インフラストラクチャが必要です。EVN は異なる仮想ネットワーク上のトラフィックのパス分離を提供します。
- ネットワークに仮想化を追加する前に、機能するキャンパス設計が確立されている必要があります。
- Virtual Routing and Forwarding (VRF) インスタンスおよびそれらを使用して、ネットワーク全体でのトラフィックの分割を維持する方法について理解する必要があります。

EVN の制約事項

- EVN トランクは、ファスト イーサネットやギガビット イーサネット、ポート チャネルなどの 802.1Q カプセル化をサポートするすべてのインターフェイスで使用できます。
- EVN トランクには追加のプラットフォームとラインカードの制約事項があります。サポートされるプラットフォームとラインカードについては、Cisco Feature Navigator、www.cisco.com/go/cfn を確認してください。
- 単一の IP インフラストラクチャを仮想化して、最大 32 のエンドツーエンドの仮想ネットワークを提供できます。
- インターフェイスに EVN トランクを設定した場合、同じインターフェイスに VRF-Lite を設定できません。
- OSPFv3 はサポートされていません。OSPFv2 はサポートされています。
- 以下は EVN でサポートされていません。
 - IS-IS
 - RIP
 - BGP ではルート レプリケーションがサポートされない
 - 特定の SNMP set 操作
- 次は、EVN トランクでサポートされていません。
 - アクセス コントロール リスト (ACL)
 - BGP インターフェイス コマンドは継承されない
 - IPv6 (vnet global を除く)

- ネットワーク アドレス変換 (NAT)
- NetFlow
- Web Cache Communication Protocol (WCCP)

EVN の概要

EVN の利点

Easy Virtual Network (EVN) は、レイヤ 3 ネットワークでのエンドツーエンドの仮想化を実現する IP ベースの仮想化テクノロジーです。ネットワーク仮想化は、ネットワークのセキュリティを保護し、また複数の仮想ネットワークに同じネットワーク インフラストラクチャを使用することで、ネットワークの費用を削減するために使用できます。それぞれ固有の論理ネットワークと一意のルーティングおよびフォワーディングテーブルを使用して、複数のグループをサポートすることにより、同じ物理インフラストラクチャを何回でも利用できます。

ネットワーク仮想化に先立って、以下の方法でもパスの分離が実現できます。

- 専用のルータを使用してパスを分離できますが、これは仮想ネットワークよりも費用がかかります。
- アクセスコントロールリスト (ACL) を使用しても実現できますが、これは一意のルーティングおよびフォワーディングテーブルをサポートしません。また、維持費用が高くなり、仮想ネットワークよりもエラーが発生しやすくなる可能性があります。

EVN には次の利点があります。

- トラフィックを分離するために個別の物理インフラストラクチャを維持する必要がないため、資本支出を削減します。トラフィック パスの分離により 1 つの IP ネットワークで複数の仮想ネットワークを設定できるため、追加のハードウェアの費用を節約します。
- 合併、買収、ビジネスパートナーのネットワーク統合が容易なため、ビジネスの柔軟性が高まります。
- ネットワークのコアを通るトラフィックの分離を維持するためのインフラストラクチャ要件が少ないため、ネットワークの複雑さを緩和します。
- Multi-VRF (VRF-Lite) と呼ばれる既存のメカニズムに基づいて構築されています。EVN は VRF-Lite と互換性があります。EVN の互換性については、「VRF-Lite」の項を参照してください。EVN はパス分離の拡張機能、簡単な設定と管理、改善された共有サービスのサポートを提供するため、VRF-Lite よりも EVN が推奨されます。

社内の部署間でトラフィックの分割を維持することに加えて、パスの分離が役に立つ他のシナリオがあります。以下に例を示します。

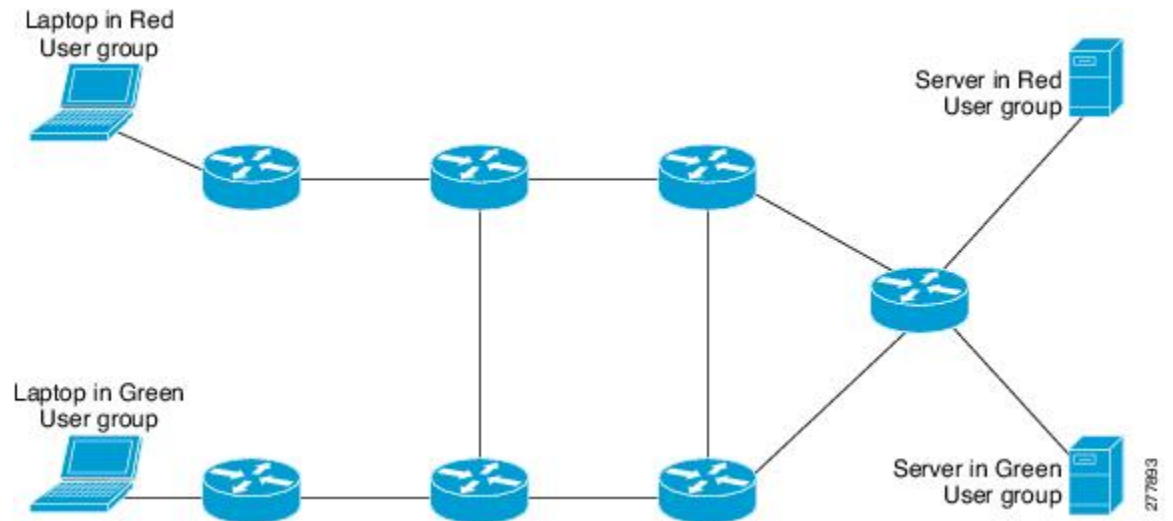
- インターネットへのゲストアクセス：お客様のネットワークを通るあらかじめ定められたデータ パスを使用し、ゲストトラフィック用に一意のデフォルト ルートを定義することで、インターネットへのゲストのネットワーク アクセスを制限します。
- ネットワーク アドミッション コントロール (NAC) の分離：非準拠のデスクトップから発信されたトラフィックを分離します。
- パートナーアクセス：インターネット、電子メール、DNS、DHCP、またはアプリケーションサーバなどのネットワーク共有サービスへのパートナーや請負業者のアクセスを制限します。
- アプリケーションとデバイスの分離：トラフィックが検査される中央のファイアウォールにトラフィックを「強制的に通す」ことによって、サービスとデバイスのセキュリティを保護します。
- アウトソーシング サービス：さまざまなクライアントからのデータ トラフィックを相互に分離します。
- スケーラブルなネットワーク：ネットワークの一部を、きわめて厳格なサービスレベルを必要とするトラフィックに制限し、それにより、必要な場所へのみそれらの要件を提供することで、コストを低減することができます。
- 子会社/合併/買収：必要に応じてサービスの共有を可能にすると同時に、段階を追って会社やネットワークを統合します。
- サービスプロバイダーとしての役割を持つ企業：自立したグループに対して1つの権限下で個別のネットワークを必要とします。例は、航空路線ごとに仮想ネットワークをサポートする空港です。

仮想ネットワーク タグによるパスの分離の提供

異なるユーザグループを同じIPインフラストラクチャで実行することは珍しくありません。さまざまなビジネス上の理由から、異なるグループ間でのトラフィックの分離が必要になります。次の図は、同じネットワークで実行する Red と Green の2つのユーザグループを示しています。ネットワーク仮想化の前は、2つのグループ間でのトラフィックの分離はありません。Red ユーザグループのユーザは、Green ユーザグループのサーバにアクセスでき、逆も可能です。

ネットワーク仮想化を使用しない場合、パスの分離はアクセス コントロールによって実現できますが、これは、維持に費用がかかり、エラーが起きやすく、ネットワークごとに一意のルーティングおよびフォワーディング テーブルをサポートしていません。

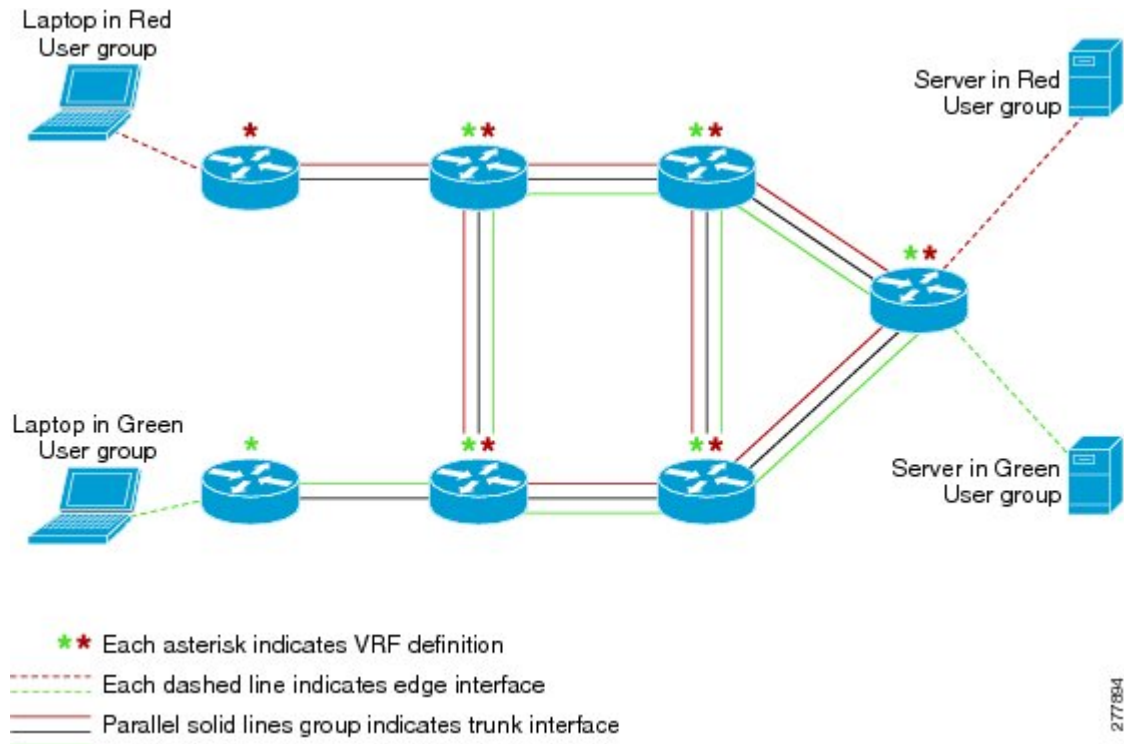
図 1: 仮想化しないネットワーク



仮想ネットワークは、1つの物理ネットワーク上のさまざまなユーザ グループを大まかにセグメント化します。仮想ネットワークを設定することによって、単一の IP インフラストラクチャを仮想化して、多数のエンドツーエンドの仮想ネットワークを実現できます。次の図では、Red と

Green の 2 つの VRF を作成して、単一の IP インフラストラクチャを 2 つの VPN に仮想化しています。

図 2: 仮想化したネットワーク



VRF を利用したデバイスレベルの分離に加えて、各仮想ネットワークは他の仮想ネットワークからパスが分離されます。トラフィックをタグ付けすることでパスの分離が実現されるため、同じ仮想ネットワーク全体では同じタグ値が伝送されます。パスに沿った各ネットワーク デバイスは、タグを使用して、異なる VRF 間で分割を行います。1 つのタグ番号は、たとえば、あるルータ上の VRF red を別のルータ上の VRF red に関連付けます。

仮想ネットワーク タグ

各 VPN と関連付けられた EVN は、設定時に割り当てるタグ値を持ちます。タグ値はグローバルです。つまり、各ルータで同じ EVN には同じ数値のタグ値が関連付けられる必要があります。タグ値の範囲は、2 ～ 4094 です。



(注)

Cisco Catalyst 6500 ファミリのネットワーク デバイスで EVN を設定する場合には、vnet tag を 2 ～ 1000 の範囲に割り当てることをお勧めします。Cisco Catalyst 6000 製品ラインの Sup2T プラットフォームでは、Cisco IOS Release 15.1(1)SY 以降、**vlan internal allocation policy descending** コマンドが設定されると、**vnet tag** の範囲が 2 ～ 3900 になります。

EVNは、ファストイーサネットやギガビットイーサネット、ポートチャネルなどの802.1Qカプセル化をサポートするすべてのインターフェイスで使用できます。VRF-Liteソリューションとの下位互換性を可能にするため、802.1qフレームのvLAN IDフィールドを使用して、仮想ネットワークタグが伝送されます。

仮想ネットワークタグを伝送するトラフィックはタグ付きトラフィックと呼ばれます。仮想ネットワークタグを伝送しないトラフィックはタグなしトラフィックと呼ばれます。

red と green の 2 つの VRF がある次の構成で、タグを説明します。

```
! Define two VRFs, red and green.
vrf definition red
  vnet tag 101
!
  address-family ipv4
  exit-address-family
!
vrf definition green
  vnet tag 102
!
  address-family ipv4
  exit-address-family
!
```

仮想ネットワークは、仮想ネットワークタグが割り当てられているVRFインスタンスとして定義されます。

vnet Global

デバイス上に「vnet global」と呼ばれる事前定義されたEVNがあります。これは、グローバルルーティングコンテキストを表し、デフォルトのRIBに対応します。図2と図3では、vnet globalがルータを接続する黒線で表されています。vnet globalはタグなしトラフィックを伝送します。デフォルトで、インターフェイスはvnet globalに属します。さらに、vnet globalは常にトランクインターフェイス上で実行します。vnet globalはデフォルトのルーティングテーブルとも呼ばれます。



(注) IPv6 トラフィックは vnet global でのみサポートされます。

エッジインターフェイスとEVN トランク インターフェイス

ユーザデバイスはレイヤ2スイッチポートに接続され、VLANに割り当てられています。VLANはレイヤ2VPNとして考えることができます。お客様は、単一のVLANの共通のレイヤ3VPNでサポートされる必要があるすべてのデバイスをグループ化します。VLANとVRF間でデータトラフィックが処理されるポイントはエッジインターフェイスと呼ばれます。

- エッジインターフェイスはユーザデバイスをEVNに接続し、事実上、EVNの境界を定義します。エッジインターフェイスはVRF対応でないホストやサーバなどのデバイスを接続します。エッジインターフェイス経由で伝送されるトラフィックはタグが付けられません。

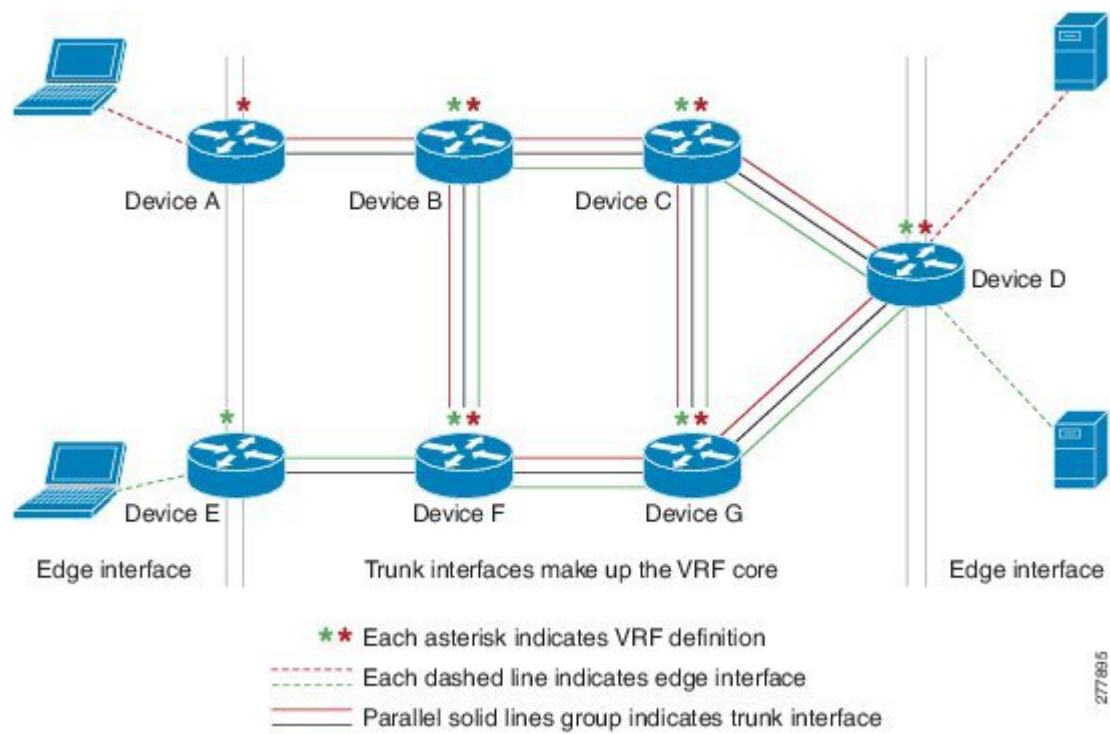
エッジインターフェイスは、受信したトラフィックが属する EVN を分類します。各エッジインターフェイスは、1 つだけの EVN に属するように設定されます。

- EVN トランク インターフェイスは、VRF 対応ルータ同士を接続し、コアに、複数の EVN のトラフィックを転送するための手段を提供します。トランク インターフェイスはタグ付きトラフィックを伝送します。タグは、対応する EVN へのパケットの多重化を解除するために使われます。トランク インターフェイスには、EVN ごとに 1 つずつサブインターフェイスがあります。インターフェイスを EVN トランク インターフェイスとして定義するには、**vnettrunk** コマンドを使用します。

EVN インターフェイスは、エッジインターフェイスとトランク インターフェイスの 2 種類のインターフェイスを使用します。インターフェイスはエッジインターフェイスまたはトランク インターフェイスのいずれかになります。両方は使用できません。図 3 には、VRF Red に属するエッジインターフェイスを持つ Router A と D を示します。Router D および E には VRF Green に属するエッジインターフェイスがあります。

Router B、C、D、F、および G には、EVN コアを作成するトランク インターフェイスがあります。これらの 5 台のルータは VRF Red と VRF Green の両方に属するインターフェイスがあります。

図 3: EVN エッジと EVN トランク インターフェイス



表示出力でのトランク インターフェイスの識別

トランク インターフェイスは複数の EVN を伝送するため、トランク インターフェイス名を表示するだけでは十分でない場合があります。表示出力がトランク インターフェイス上で実行する特定の EVN に含まれることを示す必要がある場合、使用する表記は、ピリオドと仮想ネットワーク タグを追加して、*interface.virtual-network-tag* の形式になります。例は、*gigabitethernet1/1/1.101* や *gigabitethernet1/1/1.102* のようになります。

デフォルトで、トランク インターフェイスが設定されると、すべての EVN と関連付けられた仮想ネットワーク タグが設定され、仮想ネットワーク サブインターフェイスが自動的に作成されます。上記のように、ピリオドと仮想ネットワーク タグ番号がインターフェイス番号に追加されます。

次の例では、VRF red が仮想ネットワーク タグ 3 で定義されています。したがって、システムにより、ファスト イーサネット 0/0/0.3（VRF red の）が作成されました。

```
Router# show running-config vrf red
```

```
Building configuration...
Current configuration : 1072 bytes
vrf definition red
  vnet tag 3
  !
  address-family ipv4
  exit-address-family
  !
```

この非表示インターフェイスは **showderived-config** コマンドで表示でき、ファスト イーサネット 0/0/0 に入力されたすべてのコマンドが、ファスト イーサネット 0/0/0.3 に継承されていることがわかります。

```
Router# show derived-config interface fastethernet0/0/0.3
```

```
Derived configuration : 478 bytes
!
interface FastEthernet0/0/0.3
  description Subinterface for VRF NG red
  vrf forwarding red
  encapsulation dot1Q 3
  ip address 10.1.1.1 255.255.255.0
  ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 x
  ip bandwidth-percent eigrp 1 3
  ip hello-interval eigrp 1 6
  ip hold-time eigrp 1 18
  no ip next-hop-self eigrp 1
  no ip split-horizon eigrp 1
  ip summary-address eigrp 1 10.0.0.0 255.0.0.0
end
```

トランク インターフェイスの単一の IP アドレス

トランク インターフェイスは複数の EVN にトラフィックを伝送できます。設定プロセスを簡単にするため、すべてのサブインターフェイスと関連付けられた EVN には同じ IP アドレスが割り当てられます。言い換えると、トランク インターフェイスは、異なる EVN コンテキストで、同

じ IP アドレスによって識別されます。これは、各 EVN が一意のルーティングおよびフォワーディング テーブルを持つため、複数の EVN で重複する IP アドレスをサポートできるからです。

定義されている VRF とトランク インターフェイス上で実行する VRF 間の関係

デフォルトで、ルータ上のトランク インターフェイスは、**vrf definition** コマンドで定義されたすべての VRF にトラフィックを送信します。たとえば、次の構成では、ルータに定義されたすべての VRF がインターフェイスに含まれています。

```
interface FastEthernet 1/0/0
  vnet trunk
  ip address 10.1.1.1 255.255.255.0
```

ただし、トラフィック分割の目的で、特定のトランク インターフェイス上で VRF のサブセットのみを有効にしたい場合もあります。これを実現するには、VRF リストを作成します。このリストは **vnet trunk** コマンドで参照されます。VRF リストでトランク インターフェイスを有効にすると、リスト上の VRF のみがインターフェイスで有効になります。例外として、**vnet global** はトランク インターフェイスで常に有効にされます。

次の例では、リストに指定された 2 つの VRF (red と green) のみがインターフェイスで有効にされます。

```
vrf list mylist
  member red
  member green
!
interface FastEthernet 1/0/0
  vnet trunk list mylist
  ip address 10.1.1.1 255.255.255.0
```

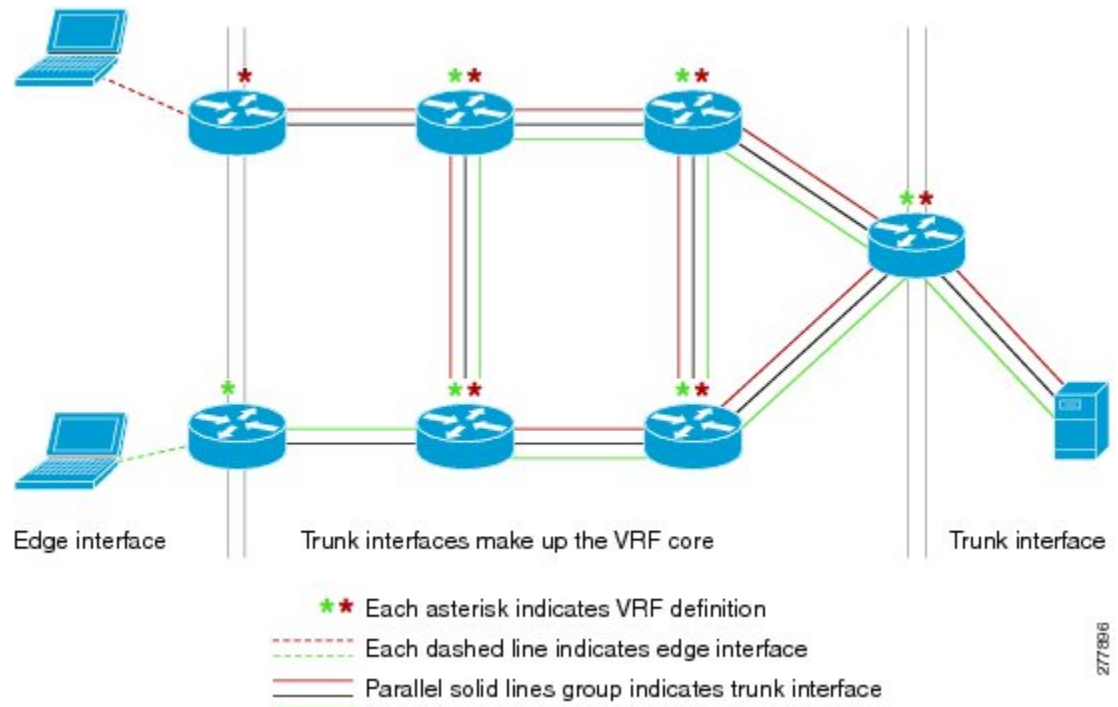
VRF 認識

仮想ネットワークに接続されたデバイスが仮想ネットワーク タグを理解できず、タグなしトラフィックしか送受信できないことがあります。そのようなデバイスは VRF 認識不能と呼ばれます。たとえば、ラップトップコンピュータは通常 VRF 認識不能です。

逆に、タグ付きトラフィックを送受信でき、そうしたトラフィックを処理する時にタグ値を考慮するデバイスは、VRF 認識と呼ばれます。たとえば、異なる EVN 間で共有されている VRF 認識サーバは、仮想ネットワークタグを使用して受信した要求を区別し、応答を送信します。VRF 認

識デバイスは、図 4 に示すように、トランク インターフェイスを使用して、EVN に接続されます。

図 4: VRF 認識サーバ



「VRF 認識」という用語は、ルータで実行するソフトウェア コンポーネントを説明するためにも使うことができます。ソフトウェア コンポーネントは異なる EVN 上で動作できる場合、VRF 認識です。たとえば、ping は、ping パケットを送信する EVN を選択できるため、VRF 認識です。

EVN でサポートされるルーティング プロトコル

各 EVN はルーティング プロトコルの個別のインスタンスを実行します。これにより、EVN ごとにルーティングを個別に微調整することが可能になり、運命共同体的な事態の発生を抑えることができます。異なる仮想ネットワークで異なるルーティング プロトコルを同時に実行できます。

EVN は、ユニキャストルーティングの場合に、スタティックルート、OSPFv2、およびEIGRPをサポートし、マルチキャストルーティングの場合に、PIM、MSDP、およびIGMPをサポートします。

仮想ネットワークのパケット フロー

パケットは、エッジインターフェイスを通過して EVN に入り、複数のトランク インターフェイスを通過して、別のエッジインターフェイスから仮想ネットワークを出ます。入力エッジインターフェイスでは、パケットが VLAN から特定の EVN にマッピングされます。パケットは、EVN に

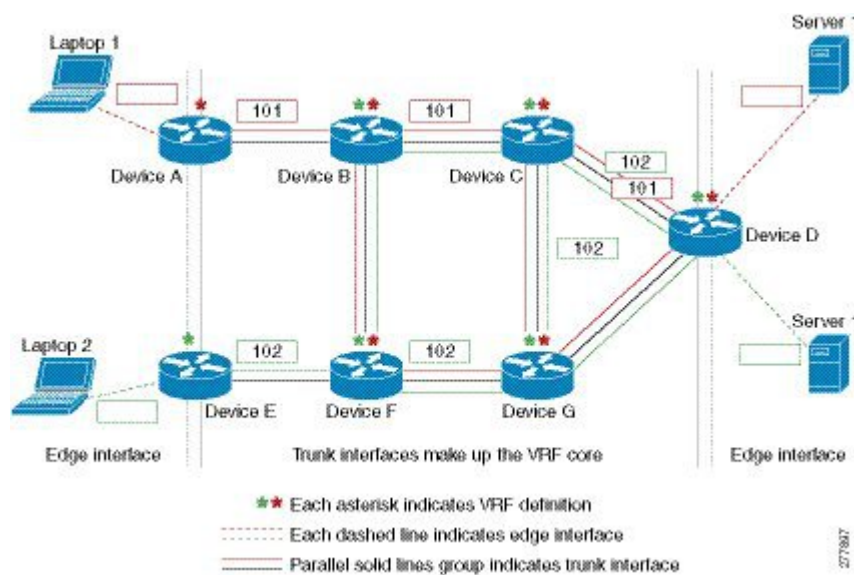
マッピングされると、関連付けられた仮想ネットワーク タグでタグ付けされます。仮想ネットワーク タグにより、トランク インターフェイスは複数の EVN にパケットを送送できます。パケットは、出力エッジ インターフェイスを通して EVN を出るまで、タグが付けられたままになります。

エッジ インターフェイスでは、インターフェイスに関連付けられている EVN がルートのルックアップに使用されます。トランク インターフェイスで、パケットで伝送された仮想ネットワーク タグが使われ、パケットをルーティングするための対応する EVN が検索されます。

出力インターフェイスがエッジインターフェイスの場合、パケットはタグが外されて転送されます。ただし、出力インターフェイスがトランク インターフェイスの場合、パケットは入力 EVN のタグが付けられて転送されます。

次の図に、red と green の 2 つの VRF からトラフィックが、タグ 101 と 102 を使用して、同じ IP インフラストラクチャ上で共存できる状況を示します。

図 5: 仮想ネットワークのパケットフロー



VRF red の Laptop 1 から Server 1 へのパケットフローは次のようになります。

- 1 Laptop 1 がタグなしパケットを Server 1 に送信します。
- 2 Router A はエッジインターフェイス上でパケットを受信し、これは VRF red に関連付けられています。
 - 1 Router A は VRF red でルートルックアップを実行し、ネクストホップがトランク インターフェイスから Router B であることがわかります。
 - 2 Router A は VRF red のタグ (101) でパケットをカプセル化し、それをトランク インターフェイス上で送信します。
- 3 Router B はトランク インターフェイス上でパケットを受信します。Router B は仮想ネットワーク タグ 101 を確認して、そのパケットが VRF red に属することを識別します。

- 1 Router B は VRF red でルートルックアップを実行し、ネクスト ホップがトランク インターフェイスから Router C であることがわかります。
- 2 Router B は VRF red のタグ (101) でパケットをカプセル化し、それをトランク インターフェイス上で送信します。
- 4 Router C はトランク インターフェイス上でパケットを受信します。Router C は仮想ネットワーク タグ 101 を使用して、そのパケットが VRF red に属することを識別します。
 - 1 Router C は VRF red でルートルックアップを実行し、ネクスト ホップがトランク インターフェイスから Router D であることがわかります。
 - 2 Router C は VRF red のタグ (101) でパケットをカプセル化し、それをトランク インターフェイス上で送信します。
- 5 Router D はトランク インターフェイス上でパケットを受信します。Router D は仮想ネットワーク タグ 101 を使用して、そのパケットが VRF red に属することを識別します。
 - 1 Router D は VRF red でルートルックアップを実行し、ネクスト ホップがエッジインターフェイスからわかります。
 - 2 Router D は、エッジインターフェイス上でタグなしパケットを Server 1 に送信します。
- 6 Server 1 は Laptop 1 から発信されたタグなしパケットを受信します。

EVN トランク インターフェイスのコマンド継承

EVN の利点の 1 つは、共通のトランク インターフェイスで複数の EVN を簡単に設定できることです。EVNに関連付けられる各インターフェイスを個別に設定する必要がありません。EVN トランク インターフェイスは、別々の EVN の設定要件が単一のトランク インターフェイス上で類似するという事実を利用します。トランク インターフェイスで特定のコマンドを設定する場合、それらは、**vnetglobal** を含め、同じインターフェイス上で実行するすべての EVN によって継承されるデフォルト値を定義します。これらの設定が、インターフェイスを共有するすべての EVN で満足できる場合は、個別の設定は必要ありません。

たとえば、OSPF hello interval は、次のように 1 行の設定で、トランク インターフェイス上のすべての EVN に設定できます。

```
interface gigabitethernet1/1/1
 vnet trunk
 ip address 10.1.2.1 255.255.255.0
 ! set OSPF hello interval for all VRFs on this interface.
 ip ospf hello-interval 20
```

同じインターフェイス上で実行するすべての EVN によって値が継承されるトランク インターフェイス上で設定されるコマンドのリストについては、「インターフェイス上の仮想ネットワークにより値が継承または上書きできるコマンド」の項にある表で説明しています。

コマンド継承のその他の例については、「*Easy Virtual Network* の設定」モジュールの設定例を参照してください。

コマンド継承仮想ネットワーク インターフェイス モードの上書き

同じトランク インターフェイス上の一部の EVN の設定を変えたい場合があります。この場合、コマンド継承に代わる方法として、仮想ネットワーク インターフェイス モードで個別の EVN に特定のコマンドを使用し、継承された値を選択的に上書きします。このモードでは、コマンドの設定が、シスコのデフォルト値またはインターフェイス コンフィギュレーション モードでのユーザ設定値を上書きします。

インターフェイス コンフィギュレーション モードで **vnetname** コマンドを入力すると、システムが仮想ネットワーク インターフェイス モードになります。このモードのシステム プロンプトは **Router(config-if-vnet)#** です。

継承された値が上書きできるコマンドのリストについては、このモジュールの「インターフェイス上の仮想ネットワークにより値が継承または上書きできるコマンド」の項にある表で説明しています。

例：コマンド継承の上書き

次の例では、VRF blue の 30 の OSPF コストでインターフェイス上の他の VRF の 20 の OSPF コストを上書きします。

```
interface gigabitethernet 2/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
 ! Set OSPF cost for all VRFs on this interface to 20.
 ip ospf cost 20
 vnet name blue
 description Subinterface for VRF NG blue
 ! Set OSPF cost for blue to 30.
 ip ospf cost 30
```

show derived コマンドで、30 のコストに変更されたサブインターフェイスが示されます。

```
Router(config-if-vnet)# do show derived | s interface GigabitEthernet2/0/0

interface GigabitEthernet2/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
 ip ospf cost 20
 interface GigabitEthernet2/0/0.200
 description Subinterface for VRF NG blue
 vrf forwarding blue
 ip address 10.1.1.1 255.255.255.0
 ip ospf cost 30
Router(config-if-vnet)#
```

例：vnet Global のみへの属性の有効化

同様に、vnet global のみに属性を有効にしたい場合があります。これを実行するには、次のように、**vnet global** インターフェイス サブモードを使用します。

```
interface gigabitethernet1/1/1
 vnet trunk
 ip address 10.1.2.1 255.255.255.0
 vnet global
```

```
! Set OSPF cost for global to 40.
ip ospf cost 40
```

この例では、ユーザは **vnet global** を除くすべての EVN に EIGRP インターフェイス属性を設定しようとしています。**vnet global** を除いて、すべての EVN は 20 秒の保留時間を継承します。**vnet global** は 20 秒の保留時間が 40 秒で上書きされます。

```
interface fastethernet 1/0/0
vnet trunk
ip address 10.1.3.1 255.255.255.0
ip hold-time eigrp 1 20
vnet global
ip hold-time eigrp 1 40
```

上書きの削除と EVN トランクから継承された値の復元

no キーワードと **default** キーワードは、それらをトランク インターフェイスに使うか、仮想ネットワーク インターフェイス モードで使うかによって、結果が異なります。ここでは、その違いについて説明します。

- トランク インターフェイス上でコマンドの前に **no** キーワードまたは **default** キーワードを入力すると、トランクがそのコマンドのシステムのデフォルト値に復元されます（これは、**no** または **default** キーワードの標準の動作結果です）。
- 仮想ネットワーク インターフェイス モードで、コマンドの前に **default** キーワードを入力すると、上書き値が削除され、トランクから継承された値が復元されます。特定の EVN の上書き値が無効になります。

次の例では、トランク インターフェイスの OSPF コストは 20 に設定されますが、VRF blue により、OSPF コストのその値が 30 に上書きされます。

```
interface gigabitethernet 2/0/0
vnet trunk
ip address 10.1.1.1 255.255.255.0
! Set OSPF cost for all VRFs on this interface to 20.
ip ospf cost 20
vnet name blue
! Set OSPF cost for blue to 30.
ip ospf cost 30
```

次のコマンドが入力された場合、OSPF コスト値が、トランク インターフェイスから継承されたコストである 20 に復元されます（20 は **ip ospf cost** コマンドのデフォルト値ではないことに注意してください）。

```
Router(config-if)# vnet name blue
Router(config-if-vnet)# default ip ospf cost
```

仮想ネットワーク インターフェイス モードでコマンドの前に **default** キーワードを入力すると、デフォルトの状態に復元されますが、**no** キーワードでは、必ずそうなるとは限りません。次の例で、**noipdampening-changeeigrp1** はダンプニングの変更を無効にします。

```
interface Ethernet1/1
vnet trunk
ip dampening-change eigrp 1 50
shutdown
vnet name red
```

```
no ip dampening-change eigrp 1
! Make sure vnet red does NOT have dampening change enabled, regardless of trunk setting.
!
```

コンフィギュレーション ファイルにコマンドの **no** 形式が表示されるかどうかの判定

機能のオンとオフを切り替える種類のコマンドの場合、コマンドの **no** 形式が設定されていると、コンフィギュレーションファイルにそれが表示されます。つまり、Nonvolatile Generation (NVGEN) は、次の例に示すように、EVN トランクからの設定を上書きします。

```
interface gigabitethernet 2/0/0
vnet trunk
ip access-group 1 in
vnet name red
no ip pim sparse-mode
no ip route-cache cef
no ip access-group in
vnet global
ip ospf cost 100
```

コマンドがその構文で **ip ospf cost cost** などの引数をとる場合、コマンドの **no** 形式によって設定が削除されますが、コンフィギュレーション ファイルには表示されません。つまり、ユーザは **ip ospf cost default-value** を入力して、より直接的な方法で継承された値を上書きできるため、NVGEN 処理されません。

EXEC コマンド ルーティング コンテキスト

複数の EXEC コマンドを発行して、単一の EVN に適用したい場合があります。複数の EXEC コマンドに VRF 名を入力する回数を減らすため、**routing-contextvrf** コマンドを使用してそれらの EXEC コマンドの VRF コンテキストを 1 回設定した後に、EXEC コマンドの使用に進みます。

次の表は、ルーティング コンテキストがない場合とある場合での 4 つの EXEC コマンドを対比しています。左の列では、各 EXEC コマンドで VRF を識別する必要があることに注意してください。右の列では、VRF コンテンツが 1 回で識別され、プロンプトがその VRF を反映して変更されるため、コマンドごとに VRF を識別する必要はありません。

表 1: EXEC コマンド ルーティング コンテキスト

ルーティング コンテキストのない EXEC コマンド	EXEC コマンド ルーティング コンテキスト
-	Router# routing-context vrf red Router%red#
Router# show ip route vrf red [VRF red のルーティング テーブル出力]	Router%red# show ip route [VRF red のルーティング テーブル出力]
Router# ping vrf red 10.1.1.1 [VRF red を使用した ping 結果]	Router%red# ping 10.1.1.1 [VRF red を使用した ping 結果]
Router# telnet 10.1.1.1 /vrf red [VRF red の 10.1.1.1 への Telnet]	Router%red# telnet 10.1.1.1 [VRF red の 10.1.1.1 への Telnet]
Router# traceroute vrf red 10.1.1.1 [VRF red の Traceroute 出力]	Router%red# traceroute 10.1.1.1 [VRF red の Traceroute 出力]

EVN の VRF-Lite との互換性

EVNはVRF-Liteと密接な互換性があります。言い換えると、外部から見て、802.1Q、SNMP MIB、およびすべての EVN インフラストラクチャが VRF-Lite とまったく同等に見えます。

次の図では、両方のルータに VRF が定義されています。左側のルータは VRF-Lite を使用し、右側のルータはタグ付きの EVN トランクを使用します。2 つの構成を図に示します。



VRF-Lite サブインターフェイスの設定と EVN トランクの設定

```
interface TenGigabitEthernet1/1/1
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
 ip pim sparse-mode
 logging event link-status
interface TenGigabitEthernet1/1/1.101
 description Subinterface for Red VRF
 encapsulation dot1Q 101
 ip vrf forwarding Red
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
```

```
interface TenGigabitEthernet 1/1/1
 vnet trunk
 ip address 10.122.5.32 255.255.255.254
 pim sparse-mode
 logging event link-status
Global Configuration:
 vrf definition red
  vnet tag 101

 vrf definition green
  vnet tag 102
```

```

ip pim sparse-mode
logging event subif-link-status
interface TenGigabitEthernet1/1/1.102
description Subinterface for Green VRF
encapsulation dot1Q 102
ip vrf forwarding Green
ip address 10.122.5.31 255.255.255.254
ip pim query-interval 333 msec
ip pim sparse-mode
logging event subif-link-status

```

マルチアドレス ファミリ VRF 構造

Cisco IOS Release 12.2(33)SB および 15.0(1)M 以前では、VRF の CLI は一度に 1 つのアドレス ファミリにのみ適用できました。たとえば、**ipvrflblue** コマンドは IPv4 アドレス ファミリにのみ適用されます。

Cisco IOS Release 12.2(33)SB および 15.0(1)M では、VRF の CLI は同じ VRF 下の複数のアドレス ファミリに適用されます。これは、マルチプロトコル VRF と呼ばれます。たとえば、**vrfdefinitionblue** コマンドは IPv4 VPN と IPv6 VPN に同時に適用されます。この場合でも、2 つのプロトコルのルーティング テーブルは異なっています。



(注) Cisco IOS XE Release 3.2S で、仮想ネットワークは、**vnet global** を除いて、IPv6 をサポートしません。

EVN での QoS 機能

Quality of Service (QoS) 設定は、EVN トランク上の主要な物理インターフェイスに適用されます。QoS ポリシーは、すべての VRF の物理インターフェイスを同時に通過するすべてのトラフィックに影響します。つまり、QoS とネットワーク仮想化は相互に独立しています。たとえば、音声用に指定された DSCP 値でマークされたトラフィックは、パケットが red VRF、blue VRF、または green VRF からの場合に音声キューに配置されます。すべての VRF のトラフィックが同じキューに配置されます。

インターフェイス上の仮想ネットワークにより値が継承または上書きできるコマンド

「EVN トランク インターフェイスのコマンド継承」の項で説明されているように、トランク インターフェイスに対して 1 度だけ定義されるインターフェイス コマンドがあり、その値は、インターフェイスを共有する各 EVN に継承されます。これらのコマンドは、トランク コマンドと呼ばれることがあります。

トランク コマンドの一部のコマンドは、仮想ネットワーク インターフェイス モードでコマンドを指定することで、値を上書きすることができます。これについては、「コマンド継承仮想インターフェイス モードの上書き」の項で説明しています。

次の表に、インターフェイス コマンドのリストと、そのコマンドがインターフェイス上の EVN により値が継承されるかどうか、および特定の EVN について上書きできるかどうかを示します。

表 2: インターフェイス上の仮想ネットワークにより継承または上書きされるインターフェイス コマンドの値

	値がインターフェイス上の EVN により継承されるかどうか	仮想ネットワークインターフェイスモードで値が上書きできるかどうか
IP コマンド		
ipaccounting	Yes	No
ipaddress	Yes	No
ipbroadcast-address	Yes	No
ipdirectedbroadcast	Yes	No
ipinformation-reply	Yes	No
ipirdp	Yes	No
ipload-sharing	Yes	No
ipmask-reply	Yes	No
ipmtu	Yes	No
ipproxy-arp	Yes	No
ipredirects	Yes	No
ipunnumbered	Yes	No
ipunreachables	Yes	No
EIGRP コマンド		
ipauthenticationkey-chaineigrp	Yes	Yes
ipauthenticationmodeeigrp	Yes	Yes
ipbandwidth-percenteigrp	Yes	Yes
ipdampening-changeeigrp	Yes	Yes
ipdampening-intervaleigrp	Yes	Yes

	値がインターフェイス上の EVN により継承されるかどうか	仮想ネットワークインターフェイスモードで値が上書きできるかどうか
iphello-intervaleigrp	Yes	Yes
iphold-timeeigrp	Yes	Yes
ipnext-hop-selfeigrp	Yes	Yes
ipsplit-horizoneigrp	Yes	Yes
ipsummary-address	Yes	Yes
EIGRP によるインターフェイスのコストの決定方法に影響するコマンド		
bandwidth (インターフェイス)	Yes	Yes
delay (インターフェイス)	Yes	Yes
OSPF コマンド		
ipospfprocess-idarea	No	Yes
ipospfauthentication	Yes	Yes
ipospfauthentication-key	Yes	Yes
ipospfbfd	Yes	Yes
ipospfcost	Yes	Yes
ipospfdatabase-filter	Yes	Yes
ipospfdead-interval	Yes	Yes
ipospfdemand-circuit	Yes	Yes
ipospfflood-reduction	Yes	Yes
ipospfhello-interval	Yes	Yes
ipospflls	Yes	Yes
ipospfmessage-digest-key	Yes	Yes
ipospfmtu-ignore	Yes	Yes

	値がインターフェイス上の EVN により継承されるかどうか	仮想ネットワークインターフェイスモードで値が上書きできるかどうか
ipospfnetwork	Yes	Yes
ipospfpriority	Yes	Yes
ipospfresync-timeout	Yes	Yes
ipospfshutdown	Yes	Yes
ipospftransmit-delay	Yes	Yes
ipospftransmit-interval	Yes	Yes
ipospfttl-security	Yes	Yes
ipospfvnetarea	No	No
IP マルチキャスト コマンド		
ipigmpaccess-group	Yes	Yes
ipigmpexplicit-tracking	Yes	Yes
ipigmphelper-address	Yes	Yes
ipigmpimmediate-leave	Yes	Yes
ipigmplast-member-query-count	Yes	Yes
ipigmplast-member-query-interval	Yes	Yes
ipigmplimit	Yes	Yes
ipigmpmroute-proxy	Yes	Yes
ipigmpproxy-service	Yes	Yes
ipigmpquerier-timeout	Yes	Yes
ipigmpquery-interval	Yes	Yes
ipigmpquery-max-response-time	Yes	Yes
ipigmpn	Yes	Yes
ipigmpunidirectional-link	Yes	Yes

■ インターフェイス上の仮想ネットワークにより値が継承または上書きできるコマンド

	値がインターフェイス上の EVN により継承されるかどうか	仮想ネットワークインターフェイスモードで値が上書きできるかどうか
ipigmpv3lite	Yes	Yes
ipigmpversion	Yes	Yes
ipmulticastboundary	Yes	Yes
ippimbidir-neighbor-filter	Yes	Yes
ippimbsr-border	Yes	Yes
ippimdense-mode	Yes	Yes
ippimdr-priority	Yes	Yes
ippimnbma-mode	Yes	Yes
ippimneighbor-filter	Yes	Yes
ippimpassive	Yes	Yes
ippimquery-interval	Yes	Yes
ippimsparse-dense-mode	Yes	Yes
ippimsparse-mode	Yes	Yes
ippimstate-refresh	Yes	Yes
マルチキャスト転送情報ベース (MFIB) コマンド		
ipmfibcef	Yes	Yes
ipmfibforwarding	Yes	Yes

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
Easy Virtual Network コマンド	『Easy Virtual Network Command Reference』
Easy Virtual Network の設定	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network の設定」モジュール
Easy Virtual Network 共有サービスとルートレプリケーションの設定	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network 共有サービスの設定」モジュール
Easy Virtual Network の管理とトラブルシューティング	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network の管理とトラブルシューティング」モジュール

MIB

MIB	MIB のリンク
<p>VRF 情報を提供する MIB は引き続き Easy Virtual Network で動作します。VRF から独立した MIB は、システムのすべての VRF に関する情報をレポートします。</p> <ul style="list-style-type: none">• CISCO-MVPN-MIB• MPLS-VPN-MIB• CISCO-VRF-MIB	<p>選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャセットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする場合、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

Easy Virtual Network の概要の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 3 : Easy Virtual Network の概要の機能情報

機能名	リリース	機能情報
EVN VNET トランク	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	Easy Virtual Network は、ネットワークのエンドツーエンドの仮想化を実現する IP ベースの仮想化テクノロジーです。単一の IP インフラストラクチャを使用して、トラフィック パスが独立した個別の仮想ネットワークを提供できます。



第 3 章

Easy Virtual Network の設定

Easy Virtual Network (EVN) は、エンドツーエンドのネットワーク仮想化を実現する IP ベースの仮想化テクノロジーです。単一の IP インフラストラクチャを使用して、トラフィック パスが相互に独立した状態で、個別の仮想ネットワークを提供できます。Easy Virtual Network を設定して、複数の仮想 IP ネットワークを設定します。

- [機能情報の確認, 27 ページ](#)
- [EVN の設定に関する前提条件, 28 ページ](#)
- [EVN の設定方法, 28 ページ](#)
- [EVN の設定例, 39 ページ](#)
- [その他の参考資料, 45 ページ](#)
- [Easy Virtual Network の設定の機能情報, 46 ページ](#)

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、[Bug Search Tool](#) およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

EVN の設定に関する前提条件

- ネットワークへの EVN の実装には、複数の仮想ネットワークの作成に使用する、単一の IP インフラストラクチャが必要です。異なる仮想ネットワーク上のトラフィックのパス分離が必要です。
- 「Easy Virtual Network の概要」モジュールの概念を理解する必要があります。
- 各ルータの EVN に属するインターフェイスを示す、ネットワーク トポロジを描くことをお勧めします。この図により、エッジインターフェイスとして設定するインターフェイスとトランク インターフェイスとして設定するインターフェイスの追跡が容易になります。

EVN の設定方法

Easy Virtual Network トランク インターフェイスの設定

このタスクを実行して、EVN トランク インターフェイスを設定します。このインターフェイスはルータを接続して、複数の仮想ネットワークにトラフィックを転送するためのコアを提供します。トランク インターフェイス経由で伝送されるトラフィックにはタグが付けられます。次のタスクは、ベース Virtual Routing and Forwarding (VRF) と、VRF red と VRF blue の 2 つの名前付き VRF があるトランク インターフェイスの設定方法を説明しています。

手順の概要

1. イネーブル化
2. `configureterminal`
3. `vrfdefinitionvrf-name`
4. `vnettagnumber`
5. `descriptionstring`
6. `address-familyipv4`
7. `exit-address-family`
8. `exit`
9. `vrfdefinitionvrf-name`
10. `vnettagnumber`
11. `descriptionstring`
12. `address-familyipv4`
13. `exit-address-family`
14. `exit`
15. `interfacetype number`
16. `ipaddressip-address mask`
17. `vnettrunk [list vrf-list-name]`
18. `vnetnamevrf-name`
19. `exit-if-vnet`
20. `noshutdown`
21. `exit`
22. `routerospf process-id`
23. `networkip-address wildcard area area-id`
24. `exit`
25. `routerospfprocess-id vrf vrf-name`
26. `networkip-address wildcard areaarea-id`
27. `exit`
28. `routerospfprocess-idvrfvrf-name`
29. `networkip-addresswildcardareaarea-id`
30. `end`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : <pre>Router> enable</pre>	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configureterminal 例 : <pre>Router# configure terminal</pre>	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 3	vrfdefinitionvrf-name 例 : <pre>Router(config)# vrf definition red</pre>	VRF ルーティング テーブル インスタンスを設定し、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	vnettagnumber 例 : <pre>Router(config-vrf)# vnet tag 100</pre>	VRF のグローバルな数値タグを指定します。 <ul style="list-style-type: none"> • 同じタグ番号を各エッジおよびトランク インターフェイス上の同じ仮想ネットワークに設定する必要があります。 • Cisco Catalyst 6500 ファミリのネットワーク デバイスで EVN を設定する場合には、vnet tag の番号を 2 ～ 1000 の範囲に割り当てることをお勧めします。
ステップ 5	descriptionstring 例 : <pre>Router(config-vrf)# description guest access</pre>	(任意) コンフィギュレーション ファイルを確認するネットワーク管理者に役立つように、VRF について記述します。
ステップ 6	address-familyipv4 例 : <pre>Router(config-vrf)# address-family ipv4</pre>	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 7	exit-address-family 例 : <pre>Router(config-vrf-af)# exit-address-family</pre>	アドレス ファミリ 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	exit 例 : <pre>Router(config-vrf)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 9	vrfdefinition <i>vrf-name</i> 例 : <pre>Router(config)# vrf definition blue</pre>	VRF ルーティング テーブル インスタンスを設定し、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 10	vnettag <i>number</i> 例 : <pre>Router(config-vrf)# vnet tag 200</pre>	VRF のグローバルな数値タグを指定します。 • 同じタグ番号を各エッジおよびトランク インターフェイス上の同じ VRF に設定する必要があります。
ステップ 11	description <i>string</i> 例 : <pre>Router(config-vrf) description Finance</pre>	(任意) コンフィギュレーション ファイルを確認するネットワーク管理者に役立つように、VRF について記述します。
ステップ 12	address-family <i>ipv4</i> 例 : <pre>Router(config-vrf) address-family ipv4</pre>	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 13	exit-address-family 例 : <pre>Router(config-vrf-af) exit-address-family</pre>	アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 14	exit 例 : <pre>Router(config-vrf)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 15	interface <i>type number</i> 例 : <pre>Router(config)# interface gigabitethernet 1/1/1</pre>	インターフェイス タイプを設定し、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 16	ipaddress <i>ip-address mask</i> 例 : <pre>Router(config-if)# ip address 10.1.1.1 255.255.255.0</pre>	インターフェイスのプライマリ IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 17	vnettrunk [list vrf-list-name] 例 : <pre>Router(config-if)# vnet trunk</pre>	トランク インターフェイスを定義します。 <ul style="list-style-type: none"> デフォルトで、vrfdefinition コマンドによって定義されたすべての VRF がルータ上のすべてのトランク インターフェイスで実行されます。そのため、VRF red と VRF blue は現在このインターフェイス上で実行しています。 list vrf-list-name コマンド要素を使用して、トランク インターフェイスで実行する VRF を制限します。
ステップ 18	vnetnamevrf-name 例 : <pre>Router(config-if)# vnet name red</pre>	(任意) 仮想ネットワーク インターフェイス モードを開始し、指定した VRF に適用する機能を設定して、グローバルの VRF 値を上書きします。 <ul style="list-style-type: none"> インターフェイス上のすべての VRF についてグローバル設定で満足できる場合は、このステップは必要ありません。 この手順の後、ipospfcost などの 1 つまたは複数の適切なコマンドを設定します (このタスクでは説明しません)。グローバル VRF 値を上書きするために使用するコマンドのリストについては、「Easy Virtual Network の概要」モジュールの表 2 を参照してください。
ステップ 19	exit-if-vnet 例 : <pre>Router(config-if-vnet) exit-if-vnet</pre>	VRF インターフェイス コンフィギュレーション モードを終了し、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 20	noshutdown 例 : <pre>Router(config-if) no shutdown</pre>	インターフェイスを再起動します。
ステップ 21	exit 例 : <pre>Router(config-if) exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 22	routerospf process-id 例 : <pre>Router(config)# router ospf 1</pre>	Open Shortest Path First (OSPF) ルーティング プロセスを設定し、それを VRF に関連付けます。 <ul style="list-style-type: none"> この OSPF インスタンスには VRF がないため、vnetglobal です。

	コマンドまたはアクション	目的
ステップ 23	network <i>ip-address wildcard area area-id</i> 例 : <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 0</pre>	OSPF を実行するインターフェイスと関連付けられたエリア ID を定義します。
ステップ 24	exit 例 : <pre>Router(config-router) exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 25	routerospf <i>process-id vrf vrf-name</i> 例 : <pre>Router(config)# router ospf 2 vrf red</pre>	OSPF ルーティング プロセスを設定し、それを VRF に関連付けます。 • 各 VRF には専用の OSPF インスタンスが必要なため、VRF ごとに異なる <i>process-id</i> を指定します。
ステップ 26	network <i>ip-address wildcard areaarea-id</i> 例 : <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 0</pre>	OSPF が実行されるインターフェイスと関連付けられたエリア ID、およびそれらのインターフェイスに対するエリア ID を定義します。
ステップ 27	exit 例 : <pre>Router(config-router) exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 28	routerospf <i>process-idvrfvrf-name</i> 例 : <pre>Router(config)# router ospf 3 vrf blue</pre>	OSPF ルーティング プロセスを設定し、それを VRF に関連付けます。 • 各 VRF には専用の OSPF インスタンスが必要なため、VRF ごとに異なる <i>process-id</i> を指定します。
ステップ 29	network <i>ip-addresswildcardareaarea-id</i> 例 : <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 2</pre>	OSPF が実行されるインターフェイスと関連付けられたエリア ID、およびそれらのインターフェイスに対するエリア ID を定義します。

	コマンドまたはアクション	目的
ステップ 30	end 例 : Router(config-vrf) end	設定セッションを終了し、特権 EXEC モードに戻ります。

トランク インターフェイス上における VRF サブセットの有効化

前のタスク「Easy Virtual Network トランク インターフェイスの設定」では、2つの VRF でトランク インターフェイスを設定する方法を示しました。デフォルトで、ルータ上のトランク インターフェイスは、**vrfdefinition** コマンドで定義された各 VRF にトラフィックを伝送できます。ただし、トランク インターフェイス上で VRF のサブセットのみを有効にしたい場合もあります。これは、VRF リストを作成して実行します。ルータには最大 32 の VRF リストが存在できます。VRF リストを作成するには、次の作業を実行します。このタスクは、VRF がすでに設定されていることを前提にしています。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **vrflistvrf-list-name**
4. **membervrf-name**
5. ステップ 4 を繰り返して、その他の VRF をリストに追加します。
6. **exit-vrf-list**
7. **interfacetypenumber**
8. **vnettrunklistvrf-list-name**
9. **ipaddressip-addressmask**
10. **end**
11. **showvrflist [vrf-list-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	イネーブル化 例 : Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configureterminal 例 : <pre>Router# configure terminal</pre>	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 3	vrflistvrf-list-name 例 : <pre>Router(config)# vrf list External</pre>	VRF のリストを定義し、VRF リスト コンフィギュレーション モードを開始します。 • <i>vrf-list-name</i> 引数には、最大 32 文字を含めることができます。疑問符、スペース、* は使用できません。
ステップ 4	membervrf-name 例 : <pre>Router(config-vrf-list)# member blue</pre>	VRF リストのメンバとして既存の VRF を指定します。 • VRF は、リストに追加する前に定義されている必要があります。
ステップ 5	ステップ 4 を繰り返して、その他の VRF をリストに追加します。	(任意) トランク インターフェイスで 1 つの VRF を有効にする場合、リストに必要な VRF は 1 つだけです。
ステップ 6	exit-vrf-list 例 : <pre>Router(config-vrf-list)# exit-vrf-list</pre>	VRF リスト コンフィギュレーション モードを終了します。
ステップ 7	interfacetypenumber 例 : <pre>Router(config)# interface gigabitethernet 1/1/1</pre>	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 8	vnettrunklistvrf-list-name 例 : <pre>Router(config-if)# vnet trunk list mylist</pre>	トランク インターフェイスを定義し、VRF リストにある VRF を有効にします。 • ステップ 3 で定義した <i>vrf-list-name</i> を使用します。
ステップ 9	ipaddressip-addressmask 例 : <pre>Router(config-if)# ip address 10.1.3.1 255.255.255.0</pre>	インターフェイスのプライマリ IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 10	end 例 : Router(config-if) end	設定セッションを終了し、特権 EXEC モードに戻ります。
ステップ 11	showvrflist [vrf-list-name] 例 : Router# show vrf list mylist	VRF リストの情報を表示します。

EVN エッジ インターフェイスの設定

次のタスクを実行して、ユーザデバイスを仮想ネットワークに接続するエッジインターフェイスを設定します。エッジインターフェイス経由で伝送されるトラフィックはタグが付けられません。エッジインターフェイスは、受信したトラフィックが属する仮想ネットワークを判断します。各エッジインターフェイスは、1 つだけの仮想ネットワークにマッピングされます。

手順の概要

1. イネーブル化
2. **configureterminal**
3. **interfacetypenumber**
4. **vrfforwardingvrf-name**
5. **ipaddressip-addressmask**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configureterminal 例 : <pre>Router# configure terminal</pre>	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 3	interfacetypenumber 例 : <pre>Router(config)# interface gigabitethernet 1/0/0</pre>	インターフェイス タイプを設定し、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	vrfforwardingvrf-name 例 : <pre>Router(config-if)# vrf forwarding red</pre>	エッジ インターフェイスを定義し、着信トラフィックが属する VRF を判断します。 <ul style="list-style-type: none"> • vrf definition コマンドで <i>vrf-name</i> がすでに定義されている必要があります。 • この例では、着信トラフィックが VRF red に属しています。 (注) エッジ インターフェイスを設定しようとする場合、トランク インターフェイス上にないことを確認してください。
ステップ 5	ipaddressip-addressmask 例 : <pre>Router(config-if)# ip address 10.1.1.1 255.255.255.0</pre>	インターフェイスのプライマリ IP アドレスを設定します。
ステップ 6	end 例 : <pre>Router(config-if) end</pre>	設定セッションを終了し、特権 EXEC モードに戻ります。

次の作業

エッジ インターフェイスとトランク インターフェイスを設定したら、ネットワーク図を参照して、別のルータにログインします。エッジ インターフェイスがある場合、そのインターフェイスを設定します。トランク インターフェイスがある場合は、適切な VRF でそのインターフェイスを設定します。各 VRF に属する各ルータおよびインターフェイスの設定を続行します。

VRF で実行する他のプロトコル機能を設定します。該当する IP ルーティング設定ガイドを参照してください。

EVN 設定の確認

このタスクの次のいずれかのステップを実行して、設定を確認します。仮想ネットワークが VRF の場合、仮想ネットワークで既存のすべての VRF **show** コマンドがサポートされます。ルータに VRF と仮想ネットワークが混在する場合、各種 **showvrf** コマンドの出力には VRF と仮想ネットワークの両方が含まれます。

手順の概要

1. イネーブル化
2. **showvnettag**
3. **showrunning-config [vrf | vnet] [vrf-name]**
4. **showvrflist [vrf-list-name]**
5. **show {vrf | vnet} [ipv4 | ipv6] [interface | brief | detail | lock] [vrf-name]**
6. **show {vrf | vnet} counters**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	showvnettag 例 : Router# show vnet tag	（任意）各タグが設定され、使われている場所を表示します。
ステップ 3	showrunning-config [vrf vnet] [vrf-name] 例 : Router# show running-config vrf green	（任意）実行コンフィギュレーションの VRF を表示し、VRF のインターフェイスを表示して、マルチ VRF のプロトコル設定を表示します。
ステップ 4	showvrflist [vrf-list-name] 例 : Router# show vrf list	（任意）各リスト内の VRF など、VRF リスト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 5	show {vrf vnet} [ipv4 ipv6] [interface brief detail lock] [vrf-name] 例 : Router# show vnet detail	(任意) VRF に関する情報を表示します。
ステップ 6	show {vrf vnet} counters 例 : Router# show vnet counters	(任意) サポートされ、設定されている VRF または仮想ネットワークの数に関する情報を表示します。

EVN の設定例

例 : network コマンドでの OSPF を使用した仮想ネットワーク

この例では、**network** コマンドで共有 VRF インターフェイスをベース VRF と red および blue の 2 つの名前付き VRF に関連付けます。各 VRF には固有の OSPF インスタンスが必要であるため、3 つの OSPF インスタンスがあります。OSPF 1 には VRF がないため、これは **vnetglobal** です。

```
vrf definition red
  vnet tag 100
  address-family ipv4
  exit-address-family
!
vrf definition blue
  vnet tag 200
  address-family ipv4
  exit-address-family
!
interface gigabitethernet 0/0/0
  ip address 10.0.0.1 255.255.255.0
  vnet trunk
  vnet name red
  ip ospf cost 100
!
router ospf 1
  log-adjacency-changes detail
  network 10.0.0.0 255.255.255.0 area 0
router ospf 2 vrf red
  log-adjacency-changes
  network 10.0.0.0 255.255.255.0 area 0
router ospf 3 vrf blue
  log-adjacency-changes
  network 10.0.0.0 255.255.255.0 area 2
```

例 : ip ospf vnet area コマンドでの OSPF を使用した仮想ネットワーク

この例では、OSPF インスタンスと特定のインターフェイス間の関連付けに関して、前の例と異なります。この例では、OSPF がトランク インターフェイスのすべての仮想ネットワークで実行しています。**ipospfvnetarea** コマンドは GigabitEthernet 0/0/0 インターフェイスと 3 つの OSPF インスタンスを関連付けます。

```
vrf definition red
vnet tag 100
address-family ipv4
exit-address-family
!
vrf definition blue
vnet tag 200
address-family ipv4
exit-address-family
!
interface gigabitethernet 0/0/0
ip address 10.0.0.1 255.255.255.0
vnet trunk
ip ospf vnet area 0
vnet name red
ip ospf cost 100
vnet name blue
ip ospf 3 area 2
!
router ospf 1
log-adjacency-changes detail
router ospf 2 vrf red
log-adjacency-changes
router ospf 3 vrf blue
log-adjacency-changes
```

例 : EIGRP 環境でのコマンド継承と仮想ネットワーク インターフェイス モードの上書き

次の例では、さまざまな EIGRP コマンドで設定された GigabitEthernet インターフェイスを示しています。

```
interface gigabitethernet0/0/0
vnet trunk
ip address 10.0.0.1 255.255.255.0
ip authentication mode eigrp 1 md5
ip authentication key-chain eigrp 1 x
ip bandwidth-percent eigrp 1 3
ip dampening-change eigrp 1 30
ip hello-interval eigrp 1 6
ip hold-time eigrp 1 18
no ip next-hop-self eigrp 1
no ip split-horizon eigrp 1
ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

トランクが設定されているため、VRF サブインターフェイスが自動的に作成され、メイン インターフェイスのコマンドが VRF サブインターフェイス (g0/0/0.3。3 は vnet tag 3 からのタグ番号です) に継承されます。

```
R1# show running-config vrf red
```

```
Building configuration...
Current configuration : 1072 bytes
vrf definition red
  vnet tag 3
  !
  address-family ipv4
  exit-address-family
  !
```

showderived-config コマンドで非表示サブインターフェイスを表示すれば、GigabitEthernet0/0/0 に入力されたすべてのコマンドが、GigabitEthernet 0/0/0.3 に継承されていることが確認できます。

```
R1# show derived-config interface gigabitethernet0/0/0.3
Building configuration...
Derived configuration : 478 bytes
!
interface GigabitEthernet0/0/0.3
  description Subinterface for VNET red
  vrf forwarding red
  encapsulation dot1Q 3
  ip address 10.0.0.1 255.255.255.0
  ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 x
  ip bandwidth-percent eigrp 1 3
  ip dampening-change eigrp 1 30
  ip hello-interval eigrp 1 6
  ip hold-time eigrp 1 18
  no ip next-hop-self eigrp 1
  no ip split-horizon eigrp 1
  ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

仮想ネットワーク インターフェイス モードを使用してそれらのコマンドを上書きできます (**vnetname** コマンドで)。次に例を示します。

```
R1(config)# interface gigabitethernet0/0/0
R1(config-if)# vnet name red
R1(config-if-vnet)# no ip authentication mode eigrp 1 md5
  ! disable authen for e0/0.3 only
R1(config-if-vnet)# ip authentication key-chain eigrp 1 y
  ! different key-chain
R1(config-if-vnet)# ip band eigrp 1 99
  ! higher bandwidth-percent
R1(config-if-vnet)# no ip dampening-change eigrp 1
  ! disable dampening-change
R1(config-if-vnet)# ip hello eigrp 1 7
R1(config-if-vnet)# ip hold eigrp 1 21
R1(config-if-vnet)# ip next-hop-self eigrp 1
  ! enable next-hop-self for e0/0.3
R1(config-if-vnet)# ip split-horizon eigrp 1
  ! enable split-horizon
R1(config-if-vnet)# no ip summary-address eigrp 1 10.0.0.1 255.0.0.0
  ! do not summarize on e0/0.3

R1(config-if-vnet)# do show running-config interface gigabitethernet0/0/0

Building configuration...
Current configuration : 731 bytes
!
interface GigabitEthernet0/0/0
  vnet trunk
  ip address 1.1.1.1 255.255.255.0
  ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 x
  ip bandwidth-percent eigrp 1 3
  ip dampening-change eigrp 1 30
  ip hello-interval eigrp 1 6
  ip hold-time eigrp 1 18
  no ip next-hop-self eigrp 1
```

例 : EIGRP 環境でのコマンド継承と仮想ネットワーク インターフェイス モードの上書き

```
no ip split-horizon eigrp 1
ip summary-address eigrp 1 1.0.0.0 255.0.0.0
vnet name red
  ip split-horizon eigrp 1
  no ip summary-address eigrp 1 1.0.0.0 255.0.0.0
  no ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 y
  ip bandwidth-percent eigrp 1 99
  no ip dampening-change eigrp 1
  ip hello-interval eigrp 1 7
  ip hold-time eigrp 1 21
  ip next-hop-self eigrp 1
!
end
```

g0/0.3 で上書き設定が使われていることに注意します。

```
R1(config-if-vnet)# do show derived-config interface g0/0.3
```

```
Building configuration...
Derived configuration : 479 bytes
!
interface GigabitEthernet0/0/0.3
description Subinterface for VNET red
vrf forwarding red
encapsulation dot1Q 3
ip address 1.1.1.1 255.255.255.0
no ip authentication mode eigrp 1 md5
ip authentication key-chain eigrp 1 y
ip bandwidth-percent eigrp 1 99
no ip dampening-change eigrp 1
ip hello-interval eigrp 1 7
ip hold-time eigrp 1 21
ip next-hop-self eigrp 1
ip split-horizon eigrp 1
no ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

vnetname サブモードで入力されたコマンドは、スティッキです。つまり、**vnetname** サブモードでコマンドを入力すると、それがデフォルト値と同じ値に設定されているかどうかに関係なく、NVGEN 処理されます。たとえばデフォルトの hello 値は 5 です。**vnetname** サブモードで **iphelloeigrp** コマンドを入力すると、それが NVGEN 処理されます。他のモードではそうなりません。

```
R1(config-if)# interface gigabitethernet0/0/2
R1(config-if)# vnet trunk
R1(config-if)# ip bandwidth-percent eigrp 1 50 <---<< this will NOT nvgen
R1(config-if)# ip hello eigrp 1 5 <---<< this will NOT nvgen
R1(config-if)# no ip authentication mode eigrp 1 md5 <---<< this will NOT nvgen
R1(config-if)# vnet name red
R1(config-if-vnet)# ip bandwidth-percent eigrp 1 50 <---<< this will nvgen
R1(config-if-vnet)# ip hello eigrp 1 5 <---<< this will nvgen
R1(config-if-vnet)# no ip authentication mode eigrp 1 md5 <---<< this will nvgen
R1(config-if-vnet)# do show running-config interface gigabitethernet0/0/2
```

```
Building configuration...
Current configuration : 104 bytes
!
interface GigabitEthernet0/0/2
vnet trunk
no ip address
vnet name red
  ip bandwidth-percent eigrp 1 50
  ip hello-interval eigrp 1 5
  no ip authentication mode eigrp 1 md5
!
```

このスティッキ要素のため、**vnetname** サブモードで設定エントリを削除するには、一般にそのコマンドのデフォルト形式を使用する必要があります。一部のコマンドは **no** 形式を使用しても削除

できますが、これは、コマンドによって異なります。**authentication** コマンドや **summary-address** コマンドなどの一部のコマンドは、**no** 形式を使用してコマンドを無効にします。

```
R1(config-if-vnet)# default ip authentication mode eigrp 1 md5
R1(config-if-vnet)# no ip bandwidth-percent eigrp 1
R1(config-if-vnet)# no ip hello eigrp 1

R1(config-if-vnet)# do show running-config interface g0/2

Building configuration...
Current configuration : 138 bytes
!
interface GigabitEthernet0/0/2
 vnet trunk
 no ip address
 vnet name red
!
end
```

例：マルチキャスト環境でのコマンド継承と仮想ネットワーク インターフェイス モードの上書き

次の例は、マルチキャストネットワークでのコマンド継承と仮想ネットワーク インターフェイス モードの上書きを示しています。トランク インターフェイスは、別々の VRF の設定要件が同じトランク インターフェイス上で類似するという事実を利用します。トランク インターフェイスに設定された適格なコマンドは、同じインターフェイス上で実行するすべての VRF に継承されます。

この例では、複数の VRF があるトランク インターフェイスに IP マルチキャスト（PIM スパースモード）が設定されています。

```
vrf definition red
 vnet tag 13
!
 address-family ipv4
 exit-address-family
!
 ip multicast-routing
 ip multicast-routing vrf red
 interface GigabitEthernet0/1/0
  vnet trunk
  ip address 125.1.15.18 255.255.255.0
  ip pim sparse-mode
```

ユーザは GigabitEthernet 0/1/0 上の VRF red に IP マルチキャストを設定しないことにしたため、仮想ネットワーク インターフェイス モードの上書きを使います。IP マルチキャストは VRF red のみに対して無効にします。**no ip pim** コマンドは、VRF red のスパースモード、デンスモード、スパース - デンスモードを含む PIM（Protocol Independent Multicast）のすべてのモードを無効にします。

```
interface GigabitEthernet0/1/0
 vnet trunk
 ip address 125.1.15.18 255.255.255.0
 ip pim sparse-mode
 vnet name red
 no ip pim
```

例：IP マルチキャストを使用した EVN

次の例では、PIM スパース モードを設定し、RP 冗長性のため、エニーキャスト RP を利用します。この例では VRF が 1 つだけ設定されています。

例では、マルチキャスト ルーティングをグローバルに、かつ各 L3 インターフェイスで有効にする方法を示しています。黒のテキストは、グローバル テーブルを設定するコマンドのグループを示し、赤のテキストは VRF red を設定するコマンドのグループを示します。

```
ip multicast-routing
interface GigabitEthernet 1/1/1
  description GigabitEthernet to core (Global)          GLOBAL TABLE
  ip pim sparse-mode
vrf definition red
  vnet tag 100
!
  address-family ipv4
  exit-address-family
!
ip multicast-routing vrf red                             VRF RED
!
interface gigabitethernet1/1/1.100
  description GigabitEthernet to core (VRF red)
  vrf forwarding red
  ip pim sparse-mode
```

エニーキャスト RP を使用して、VRF に RP を設定します。

```
interface loopback0
  description Anycast RP Global
  ip address 10.122.5.200 255.255.255.255
  ip pim sparse-mode
!
interface loopback1
  description MSDP Peering interface
  ip address 10.122.5.250 255.255.255.255          GLOBAL TABLE
  ip pim sparse-mode
!
ip msdp peer 10.122.5.251 connect-source loopback 1
ip msdp originator-id loopback 1
ip pim rp-address 10.122.5.200
access-list 10 permit 239.0.0.0 0.255.255.255
!
!
interface loopback 10
  description Anycast RP VRF Red
  vrf forwarding red
  ip address 10.122.15.200 255.255.255.255
  ip pim sparse-mode
interface loopback 11
  description MSDP Peering interface VRF red          VRF RED
  vrf forwarding red
  ip address 10.122.15.250 255.255.255.255
  ip pim sparse-mode
!
ip msdp vrf red peer 10.122.15.251 connect-source loopback 11
ip msdp vrf red originator-id loopback 11
!
ip pim vrf red rp-address 10.122.15.200
access-list 11 permit 239.192.0.0 0.0.255.255
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
Easy Virtual Network コマンド	『Easy Virtual Network Command Reference』
Easy Virtual Network 設定タスクの概要	『Easy Virtual Network Configuration Guide』の「Easy Virtual Networks の概要」モジュール
Easy Virtual Network の共有サービスとルートレプリケーションの設定タスク	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network 共有サービスの設定」モジュール
Easy Virtual Network の管理とトラブルシューティング	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network の管理とトラブルシューティング」モジュール

MIB

MIB	MIB のリンク
<p>VRF 情報を提供する MIB は引き続き EVN で動作します。VRF から独立した MIB は、システムのすべての VRF に関する情報をレポートします。</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャセットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
★枠で囲まれた Technical Assistance の場合★右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする場合、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

Easy Virtual Network の設定の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 4 : Easy Virtual Network の設定の機能情報

機能名	リリース	機能情報
EVN VNET トランク	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	<p>このモジュールでは、仮想 IP ネットワークを設定する方法について説明します。EVN は、ネットワークのエンドツーエンドの仮想化を実現する IP ベースの仮想化テクノロジーです。単一の IP インフラストラクチャを使用して、トラフィックパスが相互に独立した状態で、個別の仮想ネットワークを提供できます。</p> <p>vrfdefinition、vrfforwarding コマンドが変更されました。</p> <p>description (vrf definition サブモード)、exit-if-vnet、exit-vrf-list、member (vrf リスト)、routing-context、showrunning-configvnet、showvnet、showvnetcounters、showvnettag、showvrfcouneters、showvrflist、vnet、vnettag、vnettrunk、vrflist コマンドが導入されました。</p>
EVN OSPF	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	<p>EVN OSPF により OSPF に Easy Virtual Network のサポートが提供されます。</p> <p>ipospfdatabase-filterallout、ipospfdemand-circuit、ipospfflood-reduction、ipospfmtu-ignore、ipospfshutdown コマンドが変更されました。</p> <p>ipospfvnetarea コマンドが導入されました。</p>

機能名	リリース	機能情報
EVN EIGRP	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	EVN EIGRP により EIGRP に Easy Virtual Network のサポートが提供されます。 ipsummary-address 、 summary-metric コマンドが変更されました。
EVN マルチキャスト	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	EVN マルチキャストにより IP マルチキャストに Easy Virtual Network のサポートが提供されます。



第 4 章

EasyVirtualNetworkの管理とトラブルシューティング

このモジュールでは、Easy Virtual Network (EVN) の管理とトラブルシューティング方法について説明します。

- [機能情報の確認, 49 ページ](#)
- [EVN の管理とトラブルシューティングの前提条件, 50 ページ](#)
- [EVN の管理とトラブルシューティングの概要, 50 ページ](#)
- [EVN の管理とトラブルシューティングの方法, 51 ページ](#)
- [その他の参考資料, 56 ページ](#)
- [EVN の管理とトラブルシューティングの機能情報, 57 ページ](#)

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、[Bug Search Tool](#) およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

EVN の管理とトラブルシューティングの前提条件

- 「Easy Virtual Network の概要」と「Easy Virtual Network の設定」の項を読んで、EVN を実装してください。

EVN の管理とトラブルシューティングの概要

EXEC モードのルーティング コンテキストにより VRF の指定の繰り返しを減らす

複数の EXEC コマンドを発行して、単一の仮想ネットワークに適用したい場合があります。複数の EXEC コマンドに Virtual Routing and Forwarding (VRF) 名を入力する回数を減らすため、**routing-contextvrf** コマンドを使用して、それらの EXEC コマンドの VRF コンテキストを 1 回設定した後に、EXEC コマンドの使用に進みます。

次の表は、ルーティング コンテキストがない場合とある場合での、Cisco IOS XE ソフトウェアの 4 つの EXEC コマンドを示しています。左の列では、各 EXEC コマンドで VRF を指定する必要があることに注意してください。右の列では、VRF コンテキストが 1 回で指定され、プロンプトがその VRF を反映して変更されるため、コマンドごとに VRF を指定する必要はありません。

表 5: EXEC コマンドルーティング コンテキスト

ルーティング コンテキストのない EXEC コマンド CLI	EXEC ルーティング コンテキスト
-	Router# routing-context vrf red Router%red#
Router# show ip route vrf red [VRF red のルーティング テーブル出力]	Router%red# show ip route [VRF red のルーティング テーブル出力]
Router# ping vrf red 10.1.1.1 [VRF red を使用した ping 結果]	Router%red# ping 10.1.1.1 [VRF red を使用した ping 結果]
Router# telnet 10.1.1.1 /vrf red [VRF red の 10.1.1.1 への Telnet]	Router%red# telnet 10.1.1.1 [VRF red の 10.1.1.1 への Telnet]
Router# traceroute vrf red 10.1.1.1 [VRF red の Traceroute 出力]	Router%red# traceroute 10.1.1.1 [VRF red の Traceroute 出力]

traceroute コマンドの出力は VRF 名と VRF タグを示す

次の例に示すように、**traceroute** コマンドの出力が、着信 VRF 名/タグおよび発信 VRF 名/タグを表示するように拡張され、トラブルシューティングが簡単になりました。

```
Router# traceroute vrf red 10.0.10.12
Type escape sequence to abort.
Tracing the route to 10.0.10.12
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.13.15 (red/13,red/13) 0 msec
   10.1.16.16 (red/13,red/13) 0 msec
   10.1.13.15 (red/13,red/13) 1 msec
 2 10.1.8.13 (red/13,red/13) 0 msec
   10.1.7.13 (red/13,red/13) 0 msec
   10.1.8.13 (red/13,red/13) 0 msec
 3 10.1.2.11 (red/13,blue/10) 1 msec 0 msec 0 msec
 4 * * *
```

VRF 単位のデバッグ出力のフィルタ

EVN を使用すると、**debug condition vrf** コマンドによって VRF 単位でデバッグ出力をフィルタできます。次に、**debug condition vrf** コマンドの出力例を示します。

```
Router# debug condition vrf red

Condition 1 set
CEF filter table debugging is on
CEF filter table debugging is on
R1#
*Aug 19 23:06:38.178: vrfmgr(0) Debug: Condition 1, vrf red triggered, count 1
R1#
```

CISCO-VRF-MIB

EVN は VRF の検出と管理のための CISCO-VRF-MIB を提供します。

EVN の管理とトラブルシューティングの方法

特定の VRF への EXEC モードのルーティング コンテキストの設定

ルータに EXEC コマンドを発行する場合に Virtual Routing and Forwarding (VRF) 名を繰り返し入力する回数を減らすため、それらの EXEC コマンドのルーティング コンテキストを 1 回設定した後に、任意の順番で EXEC コマンドの入力に進みます。このタスクを実行して、EXEC モードのルーティング コンテキストを特定の VRF に設定し、次に、システムをグローバル EXEC コンテキストに戻します。

手順の概要

1. イネーブル化
2. `routing-contextvrfvrf-name`
3. `showiproute [ip-address [mask] [longer-prefixes] | protocol [process-id] | static download]`
4. `ping[protocol [tag] {host-name | system-address}]`
5. `telnethost[port]`
6. `traceroute[vrf vrf-name | topology topology-name] [protocol] destination`
7. `routing-contextvrfglobal`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : <pre>Router> enable</pre>	特権 EXEC モードをイネーブルにします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	<code>routing-contextvrfvrf-name</code> 例 : <pre>Router# routing-context vrf red</pre>	EXEC モードのルーティング コンテキストを指定した VRF にします。
ステップ 3	<code>showiproute [ip-address [mask] [longer-prefixes] protocol [process-id] static download]</code> 例 : <pre>Router%red# show ip route</pre>	（任意）ルーティング テーブルの現在の状態を表示します。 <ul style="list-style-type: none"> システム プロンプトがターゲットの VRF を反映して変更されます。 次の例に、vNET red のコンテキスト内で発行された show ip route コマンドを示します。vNET red のルーティング テーブルが表示されます。
ステップ 4	<code>ping[protocol [tag] {host-name system-address}]</code> 例 : <pre>Router%red# ping 10.1.1.1</pre>	（任意）エコー要求パケットをアドレスに送信します。 <ul style="list-style-type: none"> この例では、vNET red のコンテキスト内で発行された ping コマンドを示しています。vNET red を使用した ping 結果が表示されます。
ステップ 5	<code>telnethost[port]</code> 例 : <pre>Router%red# telnet 10.1.1.1</pre>	（任意）Telnet をサポートするホストにログインします。

	コマンドまたはアクション	目的
ステップ 6	traceroute[vrf vrf-name topology topology-name] [protocol] destination 例 : Router%red# traceroute 10.1.1.1	(任意) パケットを宛先に伝送するルートを表示します。
ステップ 7	routing-contextvrfglobal 例 : Router%red# routing-context vrf global 例 : Router>	(任意) システムをグローバル EXEC コンテキストに戻します。 • プロンプトがユーザ EXEC プロンプトに戻ります。

VRF のデバッグ出力のイネーブル化

手順の概要

1. イネーブル化
2. **debugvrf{create | delete | error | ha | initialization | interface | ipv4 | ipv6 | issu | lock | lookup | mpls | selection}**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	debugvrf{create delete error ha initialization interface ipv4 ipv6 issu lock lookup mpls selection} 例 : Router# debug vrf ipv4	VRF デバッグ情報を表示します。

仮想ネットワークの SNMP v2c コンテキストの設定

このタスクを実行して、SNMP v2c コンテキストを VRF にマッピングします。システムによって、次の SNMP v2c 設定が自動的に実行されます。

- **snmpcontext** コマンドに入力した *context-name* と同じ名前を使用したコンテキスト作成 (**snmp-servercontext** コマンドの代わり)
- **snmpcontext** コマンドに入力した *community-name* と同じ名前を使用したグループ作成 (**snmp-servergroup** コマンドの代わり)
- **snmpcontext** コマンドに入力した *community-name* と同じ名前を使用したコミュニティ作成 (**snmp-servercommunity** コマンドの代わり) デフォルトの権限は **ro** (読み取り専用) です。
- コミュニティ コンテキスト マッピング (**snmpmibcommunity-map** コマンドの代わり)

手順の概要

1. イネーブル化
2. **configureterminal**
3. **vrfdefinition** *vrf-name*
4. **address-family** **ipv4**
5. **snmpcontext***context-name* [**community***community-name* [**rw** | **ro**]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例： Router> enable	特権 EXEC モードをイネーブルにします。 • パスワードを入力します (要求された場合)。
ステップ 2	configureterminal 例： Router# configure terminal	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 3	vrfdefinition <i>vrf-name</i> 例： Router(config)# vrf definition vrf1	仮想ルーティングおよびフォワーディング (VRF) インスタンスを定義し、VRF 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	address-family ipv4 例 : Device(config-vrf)# address-family ipv4	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレスプレフィックスを使用するルーティングセッションを設定します。
ステップ 5	snmpcontext context-name [community community-name [rw ro]] 例 : Router(config-vrf)# snmp context xxx community yyy	VRF に SNMP v2c コンテキストを設定します。 • デフォルトは、読み取り専用 (ro) です。

仮想ネットワークの SNMP v3 コンテキストの設定

このタスクを実行して、SNMP v3 コンテキストを Virtual Routing and Forwarding (VRF) にマッピングします。システムによって、次の SNMP v3 設定が自動的に実行されます。

- **snmpcontext** コマンドに入力した *context-name* と同じ名前を使用したコンテキスト作成 (**snmp-servercontext** コマンドの代わり)
- グループ作成 (**snmp-servergroup** コマンドの代わり) **snmpcontext** コマンドに入力された *context-name* に「_acnf」を付けてグループ名が生成されます。
- ユーザ作成 (**snmp-serveruser** コマンドの代わり) **snmpcontext** コマンドで設定された詳細を使用してユーザが作成されます。

手順の概要

1. イネーブル化
2. **configure terminal**
3. **vrfdefinition** vrf-name
4. **address-family** ipv4
5. **snmpcontext** context-name [user username [credential | [encrypted] [auth {md5 password | sha password}]] [access {access-list-number | access-list-name | ipv6 access-list-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	イネーブル化	特権 EXEC 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
	例 : <pre>Router> enable</pre>	<ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例 : <pre>Router# configure terminal</pre>	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 3	vrfdefinition vrf-name 例 : <pre>Router(config)# vrf definition vrf1</pre>	VRF を定義し、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	address-family ipv4 例 : <pre>Device(config-vrf)# address-family ipv4</pre>	アドレス ファミリ コンフィギュレーションモードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティングセッションを設定します。
ステップ 5	snmpcontext context-name [userusername [credential [encrypted] [auth {md5 password sha password}]] [access {access-list-number access-list-name ipv6 access-list-name}]]] 例 : <pre>Router(config-vrf)# snmp context green_ctx user green_comm encrypted</pre>	VRF に SNMP v3 コンテキストを設定します。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
Easy Virtual Network コマンド	『Easy Virtual Network Command Reference』
Easy Virtual Network の概要	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network の概要」モジュール

関連項目	マニュアル タイトル
Easy Virtual Network の設定	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network の設定」モジュール
Easy Virtual Network の共有サービスとルートレプリケーション	『Easy Virtual Network Configuration Guide』の「Easy Virtual Network 共有サービス」モジュール

MIB

MIB	MIB のリンク
<p>VRF 情報を提供する MIB は引き続き Easy Virtual Network で動作します。VRF から独立した MIB は、システムのすべての VRF に関する次の情報をレポートします。</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>★枠で囲まれた Technical Assistance の場合★右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする場合、Cisco.com のログイン ID およびパスワードが必要です。</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

EVN の管理とトラブルシューティングの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを

示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェアリリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 6: EVN の管理とトラブルシューティングの機能情報

機能名	リリース	機能情報
EVN Cisco EVN MIB	Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.3SG	EVN Cisco EVN MIB により SNMP の設定が簡単になります。 snmp context コマンドが変更されました。
EVN Traceroute	Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.3SG	EVN Traceroute により、 traceroute コマンドの出力が VRF 名とタグを表示するよう に拡張されました。 traceroute コマンドが変更されました。
EVN VNET トランク	Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.3SG	ユーザは debug condition vrf コマンドを使用して、VRF 単位でデバッグ出力をフィルタできます。 debugconditionvrf 、 debugvrf コマンドが導入されました。



第 5 章

Easy Virtual Network 共有サービスの設定

この章では、Easy Virtual Network (EVN) でルート レプリケーションとルート再配布を使用してサービスを共有する方法について説明します。

- 機能情報の確認, 59 ページ
- 仮想 IP ネットワーク共有サービスの前提条件, 59 ページ
- 仮想 IP ネットワーク共有サービスの制約事項, 60 ページ
- Easy Virtual Network 共有サービスの概要, 60 ページ
- Easy Virtual Network を使用してサービスを共有する方法, 63 ページ
- Easy Virtual Network 共有サービスの設定例, 74 ページ
- その他の参考資料, 79 ページ
- Easy Virtual Network 共有サービスの機能情報, 80 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、[Bug Search Tool](#) およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

仮想 IP ネットワーク共有サービスの前提条件

- 「Easy Virtual Networks の概要」モジュールを参照してください。

- 「Easy Virtual Networks の設定」モジュールに従って EVN を実装してください。

仮想 IP ネットワーク共有サービスの制約事項

ルートレプリケーションは、スタティック、Enhanced Interior Gateway Routing Protocol (EIGRP)、および Open Shortest Path First (OSPF) ルートでサポートされています。Border Gateway Protocol (BGP) 間でルートをレプリケートすることは不可能ですが、Virtual Routing and Forwarding (VRF) 間でルートをコピーする BGP のインポートおよびエクスポート方法が仮想ネットワークで使用可能なため、問題になりません。

Easy Virtual Network 共有サービスの概要

Easy Virtual Network の共有サービス

複数の仮想ネットワークがアクセスする必要がある共通のサービス（データベースサーバやアプリケーションサーバなど）がいくつかあります。このようなサービスを共有することは、次の理由により利点があります。

- これらのサービスは通常、各グループ間で重複しません。
- 経済的、効率的で管理が容易です。
- ポリシーを中央で展開できます。

ルートの分割を実現するには、物理的または仮想的に、各仮想ネットワークに 1 つずつサービスを複製することができます。ただし、そのソリューションは、費用効果が高くないか、実現不可能である場合もあります。EVN をサポートするルータの場合、ソリューションはルートレプリケーションとルート再配布を実行することです。

ルートレプリケーションにより、ルートが仮想ネットワーク間でレプリケートされ、ある仮想ネットワークに存在するクライアントが、別の仮想ネットワークに存在するプレフィックスに到達できるため、共有サービスが可能になります。

共有サービスアプローチは、Dynamic Name Systems (DNS)、Dynamic Host Configuration Protocol (DHCP)、および企業の通信に最適です。インターネットゲートウェイへの共有アクセス用のソリューションではありません。

VRF-Lite よりも簡単な Easy Virtual Network の共有サービス

VRF-Lite の共有サービスでは、ルート識別子 (RD)、インポートおよびエクスポートによるルートターゲット、および BGP の設定が必要です。

EVN環境では、簡単な展開であるルートレプリケーションによって、ルート共有サービスを実現できます。ルートレプリケーションではBGP、RD、ルートターゲット、インポートやエクスポートも必要ありません。

要約すると、VRF間でルートをコピーするためのBGPインポートおよびエクスポート方法はRF-LiteとEVNの両方で機能します。ただし、ルートレプリケーションは、複数の仮想ネットワークで共通サービスの共有を可能にするより簡単な代替方法です。

Easy Virtual Network のルート レプリケーション プロセス

共有サービスでは、クライアントとサーバが異なる仮想ネットワークに存在します。クライアントとサーバ間の接続を実現するには、仮想ネットワーク間でルートが交換される必要があります。VRF-LiteとEVNのどちらが実装されているかに従い、VRF間のルート交換が以下のいずれかの方法で実現されます。

- VRF-Lite が実装されている場合、ルート インポート/エクスポート機能を使用して、BGP によってルート リークが実現されます。
- EVN が実装されている場合、ルーティング情報ベース（RIB）によってルート レプリケーションが直接サポートされます。BGPには依存しません。ルートが異なる仮想ネットワークからレプリケートされると、Interior Gateway Protocol（IGP）への既存の再配布により、それらのルートが各仮想ネットワーク全体に伝播されます。

次のルートレプリケーションシナリオでは、ルータに Services と User-A という名前の2つのVRFがあります。OSPFは次のように設定されます。

```
router ospf 99 vrf services
 network 126.1.0.0 0.0.255.255 area 0
!
router ospf 98 vrf user-a
 network 126.1.0.0 0.0.255.255 area 0
```

さらに、VRF User-A にルート再配布が設定されます。

```
vrf definition user-a
!
 address-family ipv4
  route-replicate from vrf services unicast ospf 99
 exit-address-family
```

このシナリオでは、VRFサービスの次のRIBに4つのルートが含まれ、それらのうち3つはVRF User-AのRIBにレプリケートされます。ルートレプリケーションにより、次の図のようにソースRIBへのリンクが作成されます。

RIB—VRF Services

Route	Type	Destination Interface	Next Hop
126.1.17.0/24	Connected	Gi0/1	
126.1.9.0/24	OSPF	Gi0/1	126.1.17.13
126.1.12.0/24	OSPF	Gi0/1	126.1.17.13
126.1.14.0/24	OSPF	Gi0/1	126.1.17.13

RIB—VRF User-A

Route	Type	Destination Interface	Next Hop
126.1.9.0/24	OSPF	Gi0/1	126.1.17.13
126.1.12.0/24	OSPF	Gi0/1	126.1.17.13
126.1.14.0/24	OSPF	Gi0/1	126.1.17.13

ルート レプリケーションの設定により、仮想 IP ネットワーク間の相互の再配布が可能になります。共有サービスの場合、共有サービスにアクセスする必要がある VRF 内にルート レプリケーションを設定します。各 **route-replicate** コマンド内で、ルーティング ループを避けるため、ルート マップを使用して、任意でルートをフィルタで除外できます。つまり、ルートを元のルーティング プロトコルに再配布しないようにできます。ネイティブルートをレプリケートされたルートとして表示しないようにできます。

ルート レプリケーションを実装する場所

ルート レプリケーションは、できるだけ共有サービスの近くのルータに実装することをお勧めします。理想的には、サーバサブネットに直接接続されているルータを使用して、サーバ VRF 上のホストプレフィックスを再配布する必要をなくし、それにより、ルーティング ループの可能性を回避する必要があります。

Easy Virtual Network のルート レプリケーションの動作

この項では、EVNでのルート レプリケーションの動作について説明します。これはマルチトポロジルーティングでの動作とは異なります。EVN 環境では、

- **route-replicate** コマンドは **address-family ipv4** コマンドでのみ受け入れられ、**vrf definition** コマンドで設定されます。
- **route-replicate** コマンドは、指定されたアドレス ファミリ内のベース トポロジにルートをレプリケートします。
- ソース プロトコルとして **all** を指定した場合、指定した宛先トポロジに対して、VRF あたり 1 つの **route-replicate** コマンドのみが許可されます。

- **no route-replicate** コマンドは、ソース プロトコルを除外できます。
- ソース プロトコルとして **all** を指定した場合、接続されているルートがレプリケートされます (**route-replicate** コマンドのマルチトポロジ ルーティング バージョンとは異なります)。
- レプリケートされたルートは、ソースルートのアドミニストレーティブディスタンスとソース プロトコルを継承します。

Easy Virtual Network でのルート レプリケーション後のルート プリファレンス ルール

ルートがレプリケートされる場合、次のルールによってルート プリファレンスが決定します。

- 2つのルートが同じプロトコルで所有されていて、それらのソース VRF が同じであり、一方のルートがレプリケートされていない場合、レプリケートされていないルートが優先されます。

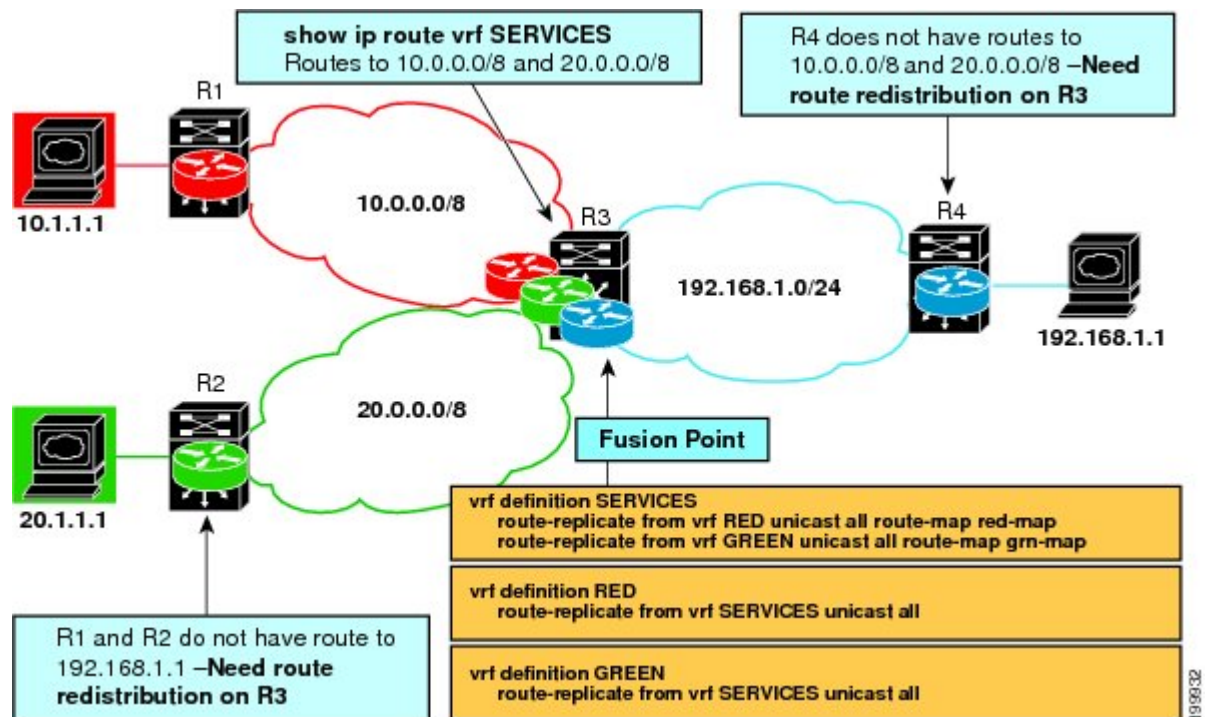
上のルールが適用しない場合、次のルールによって、この順番でルート プリファレンスが決定されます。

- 1 アドミニストレーティブ ディスタンスが小さいルートが優先されます。
- 2 デフォルトのアドミニストレーティブ ディスタンスが小さいルートが優先されます。
- 3 レプリケートされたルートよりもレプリケートされていないルートが優先されます。
- 4 元の vrf 名を比較します。辞書編集上で vrf 名が小さいルートが優先されます。
- 5 元のサブアドレスファミリを比較します。マルチキャストよりユニキャストが優先されます。
- 6 最も古いルートが優先されます。

Easy Virtual Network を使用してサービスを共有する方法

Easy Virtual Network でサービスを共有するためのルート レプリケーションの設定

次のタスクを実行して、ある VRF から別の VRF にルートをレプリケートします。このタスクテーブル内の例は次の図に基づいています。



このタスクでは、VRF SERVICES からのルートが VRF RED と VRF GREEN の両方にレプリケートされ、VRF RED と VRF GREEN 間でのルートの共有は許可されません。双方向トラフィックを許可するため、VRF RED と VRF GREEN からのルートも VRF SERVICES にレプリケートされます。



(注) 実際の EVN 環境では、VRF SERVICES と 3 つ目の VRF 間、およびもっと多くの VRF 間のルート レプリケーションが存在することがあります。そうしたレプリケーションは、簡潔さのため、次の設定タスクから外しています。

手順の概要

1. **イネーブル化**
2. **configureterminal**
3. **vrfdefinition***vrf-name*
4. **vnettag***number*
5. **description***string*
6. **address-family***ipv4*
7. **exit**
8. **exit**
9. **vrfdefinition***vrf-name*
10. **vnettag***number*
11. **description***string*
12. **address-family***ipv4*
13. **exit**
14. **exit**
15. **interface***type**number*
16. **vrf***forwarding**vrf-name*
17. **ip***address**ip-address**mask*
18. **noshutdown**
19. **exit**
20. **routerospf***process-id**vrf**vrf-name*
21. **network***ip-address**wildcard-mask**area**area-id*
22. **exit**
23. **routerospf***process-id* [*vrf**vrf-name*]
24. **network***ip-address**wildcard-mask**area**area-id*
25. **exit**
26. **vrfdefinition***vrf-name*
27. **address-family***ipv4*
28. **route-replicate***from* [*vrf**vrf-name*] {*multicast*|*unicast*} {*all*|*protocol-name*} [*route-map**map-tag*]
29. **exit**
30. **exit**
31. **vrfdefinition***vrf-name*
32. **address-family***ipv4*
33. **route-replicate***from* [*vrf**vrf-name*] {*multicast*|*unicast*} {*all*|*protocol-name*} [*route-map**map-tag*]
34. **end**
35. **show***ip**route**ev**vrf**vrf-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	イネーブル化 例 : <pre>Router> enable</pre>	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	configureterminal 例 : <pre>Router# configure terminal</pre>	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 3	vrfdefinitionvrf-name 例 : <pre>Router(config)# vrf definition SERVICES</pre>	VRF を定義し、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	vnettagnumber 例 : <pre>Router(config-vrf)# vnet tag 100</pre>	VRF のグローバルな数値タグを指定します。 • 同じタグ番号を各エッジおよびトランク インターフェイス上の同じ VRF に設定する必要があります。
ステップ 5	descriptionstring 例 : <pre>Router(config-vrf)# description shared services</pre>	（任意）コンフィギュレーション ファイルを参照するネットワーク管理者に役立つように、VRF について記述します。
ステップ 6	address-familyipv4 例 : <pre>Router(config-vrf)# address-family ipv4</pre>	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 7	exit 例 : <pre>Router(config-vrf-af)# exit</pre>	VRF 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	exit 例 : <pre>Router(config-vrf)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 9	vrfdefinition <i>vrf-name</i> 例 : Router(config)# vrf definition RED	VRF を定義し、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 10	vnettag <i>number</i> 例 : Router(config-vrf)# vnet tag 200	VRF のグローバルな数値タグを指定します。 • 同じタグ番号を各エッジおよびトランク インターフェイス上の同じ VRF に設定する必要があります。
ステップ 11	description <i>string</i> 例 : Router(config-vrf)# description user of services	(任意) コンフィギュレーション ファイルを参照するネットワーク管理者に役立つように、VRF について記述します。
ステップ 12	address-family <i>ipv4</i> 例 : Router(config-vrf)# address-family ipv4	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 13	exit 例 : Router(config-vrf-af)# exit	VRF コンフィギュレーション モードを終了します。
ステップ 14	exit 例 : Router(config-vrf)# exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 15	interface <i>type</i> <i>number</i> 例 : Router(config)# interface gigabitethernet 0/0/0	インターフェイスのタイプと番号を設定し、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 16	vrf forwarding <i>vrf-name</i> 例 : Router(config-if)# vrf forwarding SERVICES	インターフェイスと VRF インスタンスを関連付けます。

	コマンドまたはアクション	目的
ステップ 17	ipaddressip-addressmask 例 : <pre>Router(config-if)# ip address 192.168.1.3 255.255.255.0</pre>	インターフェイスのプライマリ IP アドレスを設定します。
ステップ 18	noshutdown 例 : <pre>Router(config-if)# no shutdown</pre>	インターフェイスを再起動します。
ステップ 19	exit 例 : <pre>Router(config-if)# exit</pre>	グローバルコンフィギュレーションモードに戻ります。
ステップ 20	routerospfprocess-idvrfvrf-name 例 : <pre>Router(config)# router ospf 99 vrf SERVICES</pre>	OSPF ルーティングプロセスを設定し、ルータ 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 OSPF を使用していますが、EIGRP も使用できます。
ステップ 21	networkkip-addresswildcard-maskareaarea-id 例 : <pre>Router(config-router)# network 192.168.1.0 0.0.0.255 area 0</pre>	OSPF が実行するインターフェイスと、それらのインターフェイスに対するエリア ID を定義します。
ステップ 22	exit 例 : <pre>Router(config-router)# exit</pre>	グローバルコンフィギュレーションモードに戻ります。
ステップ 23	routerospfprocess-id [vrfvrf-name] 例 : <pre>Router(config)# router ospf 98 vrf RED</pre>	OSPF ルーティングプロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 24	networkkip-addresswildcard-maskareaarea-id 例 : <pre>Router(config-router)# network 192.168.1.0 0.0.0.255 area 0</pre>	OSPF が実行するインターフェイスと、それらのインターフェイスに対するエリア ID を定義します。

	コマンドまたはアクション	目的
ステップ 25	exit 例 : <pre>Router(config-router)# exit</pre>	現在のモードを終了して、グローバルコンフィギュレーション モードに戻ります。
ステップ 26	vrfdefinition <i>vrf-name</i> 例 : <pre>Router(config)# vrf definition RED</pre>	VRF を定義し、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 27	address-family <i>ipv4</i> 例 : <pre>Router(config-vrf)# address-family ipv4</pre>	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 28	route-replicate <i>from[vrfvrf-name] {multicast unicast} {all protocol-name} [route-mapmap-tag]</i> 例 : <pre>Router(config-vrf-af)# route replicate from vrf SERVICES unicast all</pre>	指定したアドレス ファミリ内のベース トポロジにルートをレプリケートします。 <ul style="list-style-type: none"> ソース プロトコルとして all キーワードを指定した場合、指定した宛先トポロジに対して、VRF あたり 1 つの route-replicate コマンドのみが許可されます。 接続されたルートのみをレプリケートするには、ソース <i>protocol-name</i> として connected キーワードを使用します。
ステップ 29	exit 例 : <pre>Router(config-vrf-af)# exit</pre>	VRF コンフィギュレーション モードを終了します。
ステップ 30	exit 例 : <pre>Router(config-vrf)# exit</pre>	グローバルコンフィギュレーションモードに戻ります。
ステップ 31	vrfdefinition <i>vrf-name</i> 例 : <pre>Router(config)# vrf definition SERVICES</pre>	VRF を定義し、VRF 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 32	address-family ipv4 例 : <pre>Router(config-vrf)# address-family ipv4</pre>	アドレス ファミリ コンフィギュレーション モードを開始し、標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 33	route-replicate from [vrf vrf-name] {multicast unicast} {all protocol-name} [route-map map-tag] 例 : <pre>Router(config-vrf-af)# route replicate from vrf RED unicast all</pre>	指定したアドレス ファミリ内のベース トポロジにルートをレプリケートします。 • これは、双方向トラフィックを許可するステップ 28 と逆のレプリケーションです。
ステップ 34	end 例 : <pre>Router(config-vrf-af)# end</pre>	設定モードを終了します。
ステップ 35	show ip route vrf vrf-name 例 : <pre>Router# show ip route vrf RED</pre>	(任意) プラス記号 (+) で示されているレプリケートされたルートを含めて、ルートを表示します。

例

次に、前記のタスク テーブルのタスクに基づいた **show ip route vrf** コマンドの出力例を示します。

```
Router# show ip route vrf RED

Routing Table: RED
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override
Gateway of last resort is not set
  192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    +   192.168.1.0/24 is directly connected (SERVICES), GigabitEthernet0/0/0
L    +   192.168.1.3/32 is directly connected (SERVICES), GigabitEthernet0/0/0
Router#
```

次の作業

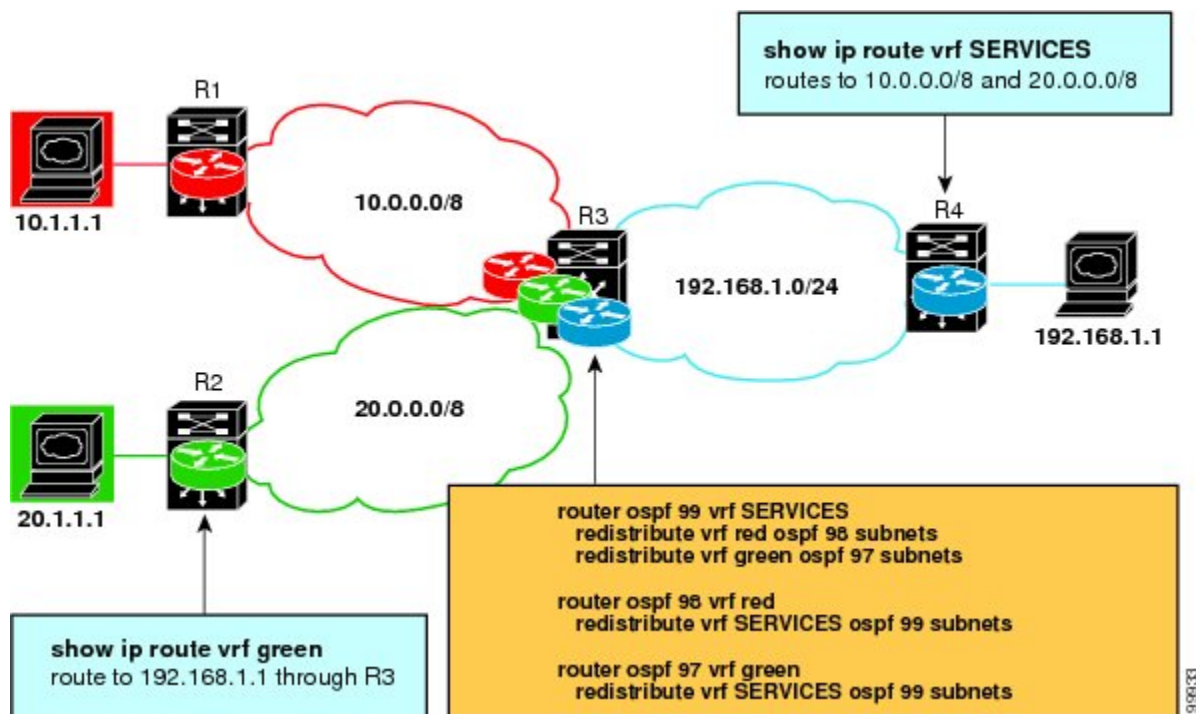
「Easy Virtual Network でサービスを共有するためのルート レプリケーションの設定」作業を実行後、Router 3 には 10.0.0.0/8 および 20.0.0.0/8 へのルートが設定され、Router 1 および Router 2 には 192.168.1.0/24 へのルートが設定されていることに注意して、上記の図のとおり VRF GREEN を設定する必要があります。

設定が完了しても、Router 1 と Router 2 にはまだ 192.168.1.1 に存在する共有サービスへのルートが設定されておらず、また Router 4 は 10.0.0.0/8 および 20.0.0.0/8 へのルートが設定されていません。それらのアクセスには、次の「EVN の共有サービスへの再配布の設定」タスクで実行するルート再配布が必要です。

Easy Virtual Network の共有サービスへの再配布の設定

このタスクは、EVN の共有サービスにルート レプリケーションを設定するタスクが実行済みであるという仮定に基づいています。

次の図は、上記の図で使ったのと同じネットワークを示しています。このタスクでは、Router 1 と Router 2 に 192.168.1.1 に存在する共有サービスへのルートが設定されるように、Router 3 で再配布を実行します。



手順の概要

1. イネーブル化
2. **configureterminal**
3. **routerospfprocess-idvrfvrf-name**
4. **redistributevrfvrf-nameospfprocess-id subnets**
5. **redistributevrfvrf-nameospfprocess-idsubnets**
6. **exit**
7. **routerospfprocess-idvrfvrf-name**
8. **redistributevrfvrf-nameospfprocess-idsubnets**
9. **exit**
10. **routerospfprocess-idvrfvrf-name**
11. **redistributevrfvrf-nameospfprocess-idsubnets**
12. **end**
13. **showiproutevrfvrf-name**

手順の詳細

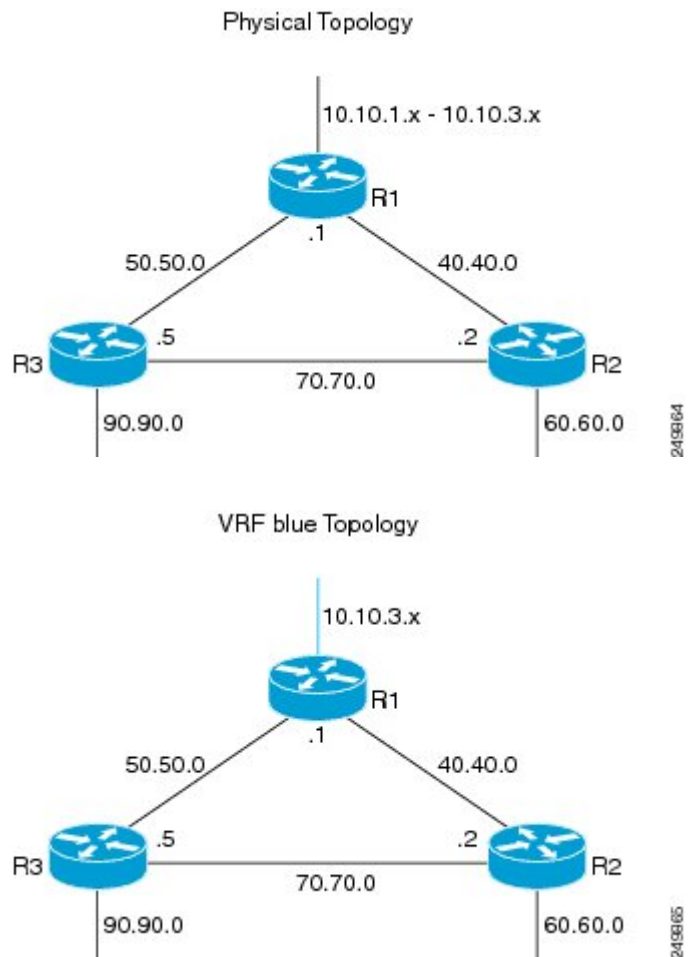
	コマンドまたはアクション	目的
ステップ 1	イネーブル化 例 : <pre>Router> enable</pre>	特権 EXEC モードをイネーブルにします。 ・パスワードを入力します（要求された場合）。
ステップ 2	configureterminal 例 : <pre>Router# configure terminal</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	routerospfprocess-idvrfvrf-name 例 : <pre>Router(config)# router ospf 99 vrf SERVICES</pre>	OSPF ルーティング プロセスを設定し、ルータ 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	redistributevrfvrf-nameospfprocess-id subnets 例 : <pre>Router(config-router)# redistribute vrf RED ospf 98 subnets</pre>	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。

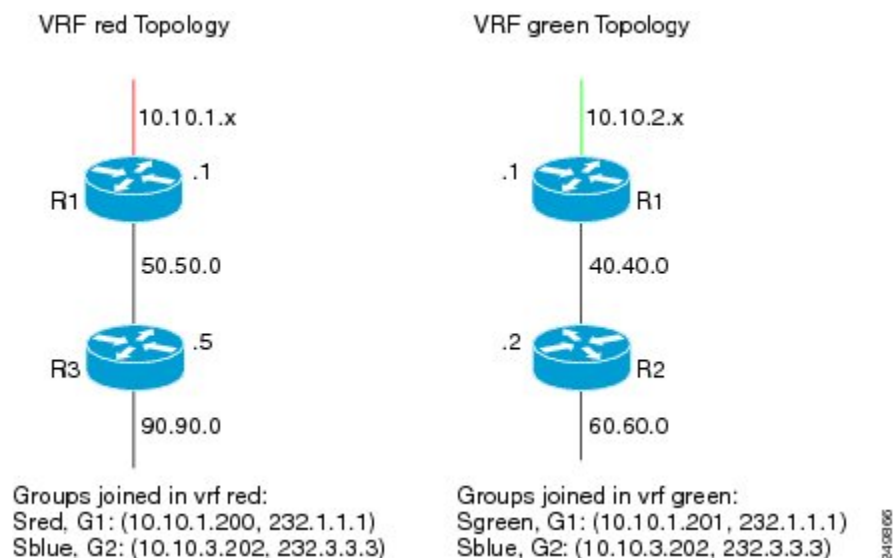
	コマンドまたはアクション	目的
ステップ 5	redistribute vrf vrf-name ospf process-id subnets 例 : <pre>Router(config-router)# redistribute vrf GREEN ospf 97 subnets</pre>	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 6	exit 例 : <pre>Router(config-router)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	router ospf process-id vrf vrf-name 例 : <pre>Router(config)# router ospf 98 vrf RED</pre>	OSPF ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 8	redistribute vrf vrf-name ospf process-id subnets 例 : <pre>Router(config-router)# redistribute vrf SERVICES ospf 99 subnets</pre>	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 9	exit 例 : <pre>Router(config-router)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 10	router ospf process-id vrf vrf-name 例 : <pre>Router(config)# router ospf 97 vrf GREEN</pre>	OSPF ルーティング プロセスを設定し、ルータ 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	redistribute vrf vrf-name ospf process-id subnets 例 : <pre>Router(config-router)# redistribute vrf SERVICES ospf 99 subnets</pre>	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
ステップ 12	end 例 : <pre>Router(config-router)# end</pre>	設定モードを終了します。

	コマンドまたはアクション	目的
ステップ 13	show ip route vrf vrf-name 例 : Router# show ip route vrf RED	(任意) プラス記号 (+) で示されているレプリケートされたルートを含めて、ルートを表示します。

Easy Virtual Network 共有サービスの設定例

例：マルチキャスト環境での **Easy Virtual Network** ルート レプリケーションとルート再配布





上記の図には、以下の3つのマルチキャストストリームがあります。

- Sred、G1：(10.10.1.200、232.1.1.1)：VRF red のソースとレシーバ
- Sgreen、G1：(10.10.2.201、232.1.1.1)：VRF green のソースとレシーバ
- Sblue、G2：(10.10.3.202、232.3.3.3)：blue のソースと、VRF red と green のレシーバ

VRF blue (10.10.3.0/24) のサーバプレフィックスは、R3 と R2 の VRF にレプリケートされ、配布されます。

マルチキャストグループ 232.3.3.3 と VRF blue のそのソースは、VRF red と VRF green の両方にレシーバがあります。ストリームは共有 VRF (blue) 経由で送信され、R3 の VRF と R2 の VRF green にレプリケートされます。

R1 の設定

```
vrf definition blue
vnet tag 4
!
address-family ipv4
exit-address-family
!
vrf definition green
vnet tag 3
!
address-family ipv4
exit-address-family
!
vrf definition red
vnet tag 2
!
address-family ipv4
exit-address-family
!
vrf list vnet-list1
member blue
member red
!
vrf list vnet-list2
```

```

    member blue
    member green
    !
vrf list vnet-list3
    member blue
    !
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf green distributed
ip multicast-routing vrf blue distributed
    !
interface FastEthernet0/0/2
    vnet trunk list vnet-list1                                [vnet trunk for red and blue]
    ip address 50.50.0.1 255.255.255.0
    no ip redirects
    no ip proxy-arp
    ip pim sparse-dense-mode
    !
interface GigabitEthernet0/1/1
    vnet trunk list vnet-list2                                [vnet trunk for green and blue]

    ip address 40.40.0.1 255.255.255.0
    no ip redirects
    no ip proxy-arp
    ip pim sparse-dense-mode
    !
interface GigabitEthernet0/1/3
    ip address 10.10.0.1 255.255.255.0
    no ip redirects
    no ip proxy-arp
    ip pim sparse-dense-mode
    !
interface GigabitEthernet0/1/3.2
    vrf forwarding red
    encapsulation dot1Q 2
    ip address 10.10.1.1 255.255.255.0
    ip pim sparse-dense-mode
    !
interface GigabitEthernet0/1/3.3
    vrf forwarding green
    encapsulation dot1Q 3
    ip address 10.10.2.1 255.255.255.0
    ip pim sparse-dense-mode
    !
interface GigabitEthernet0/1/3.4
    vrf forwarding blue
    encapsulation dot1Q 4
    ip address 10.10.3.1 255.255.255.0
    ip pim sparse-dense-mode
    !
router ospf 201 vrf red
    nsf
    redistribute connected subnets
    network 10.10.1.0 0.0.0.255 area 0
    network 50.50.0.0 0.0.0.255 area 0
    !
router ospf 202 vrf green
    nsf
    network 10.10.2.0 0.0.0.255 area 0
    network 40.40.0.0 0.0.0.255 area 0
    !
router ospf 203 vrf blue
    router-id 11.11.11.11
    nsf
    network 10.10.3.0 0.0.0.255 area 0
    network 40.40.0.0 0.0.0.255 area 0
    network 50.50.0.0 0.0.0.255 area 0
    !
router ospf 200
    nsf
    redistribute connected subnets
    network 10.10.0.0 0.0.0.255 area 0
    network 40.40.0.0 0.0.0.255 area 0

```

```

network 50.50.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!

```

R2 の設定

```

vrf definition blue
vnet tag 4
!
address-family ipv4
exit-address-family
!
vrf definition green
vnet tag 3
!
address-family ipv4
route-replicate from vrf blue unicast all route-map blue-map
[replicate routes from blue to green]
exit-address-family
!
vrf definition red
vnet tag 2
!
address-family ipv4
exit-address-family
!
vrf list vnet-list1
member blue
member green
!
vrf list vnet-list2
member blue
!
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf green distributed
ip multicast-routing vrf blue distributed
!
interface FastEthernet0/0/6
vnet trunk list vnet-list2 [vnet trunk for blue]
ip address 70.70.0.2 255.255.255.0
no ip redirects
no ip proxy-arp
ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/2
vnet trunk list vnet-list1 [vnet trunk for green and blue]
ip address 40.40.0.2 255.255.255.0
no ip redirects
no ip proxy-arp
ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/4
vnet trunk list vnet-list1 [vnet trunk for green and blue]

ip address 60.60.0.2 255.255.255.0
no ip redirects
no ip proxy-arp
ip pim sparse-dense-mode
!
router ospf 202 vrf green
redistribute connected subnets
redistribute vrf blue ospf 203 subnets route-map blue-map [redistribute routes replicated
from blue in red]
network 40.40.0.0 0.0.0.255 area 0
network 60.60.0.0 0.0.0.255 area 0
!

```

```

router ospf 203 vrf blue
router-id 22.22.22.22
network 40.40.0.0 0.0.0.255 area 0
network 60.60.0.0 0.0.0.255 area 0
network 70.70.0.0 0.0.0.255 area 0
!
router ospf 200
redistribute connected subnets
network 40.40.0.0 0.0.0.255 area 0
network 60.60.0.0 0.0.0.255 area 0
network 70.70.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!
ip prefix-list server-prefix seq 5 permit 10.10.3.0/24
!
route-map blue-map permit 10
match ip address prefix-list server-prefix
!

```

R3 の設定

```

vrf definition blue
vnet tag 4
!
address-family ipv4
exit-address-family
!
vrf definition green
vnet tag 3
!
address-family ipv4
exit-address-family
!
vrf definition red
vnet tag 2
!
address-family ipv4
route-replicate from vrf blue unicast all route-map blue-map [replicate routes from
blue to red]
exit-address-family
!
vrf list vnet-list1
member blue
member red
!
vrf list vnet-list2
member blue
!
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf green distributed
ip multicast-routing vrf blue distributed
!
interface GigabitEthernet0/2/0
vnet trunk list vnet-list1 [vnet trunk for red and blue]
ip address 90.90.0.5 255.255.255.0
no ip redirects
no ip proxy-arp
ip pim sparse-dense-mode
!
interface GigabitEthernet1/2/0
vnet trunk list vnet-list1 [vnet trunk for red and blue]
ip address 50.50.0.5 255.255.255.0
no ip redirects
no ip proxy-arp
ip pim sparse-dense-mode
!

```

```

interface FastEthernet2/0/0
 vnet trunk list vnet-list2                                [vnet trunk for blue]
 ip address 70.70.0.5 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
router ospf 201 vrf red
 redistribute connected subnets
 redistribute vrf blue ospf 203 subnets route-map blue-map    [redistribute routes
 replicated from blue in red]
 network 50.50.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
!
router ospf 203 vrf blue
 router-id 55.55.55.55
 network 50.50.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
!
router ospf 200
 redistribute connected subnets
 network 50.50.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!
ip prefix-list server-prefix seq 5 permit 10.10.3.0/24
!
route-map blue-map permit 10
 match ip address prefix-list server-prefix
!

```

その他の参考資料

関連資料

関連項目	マニュアルタイトル
Cisco IOS コマンド	『 Cisco IOS Master Command List, All Releases 』
Easy Virtual Network コマンド	『 Easy Virtual Network Command Reference 』
Easy Virtual Network の概要	『 <i>Easy Virtual Network Configuration Guide</i> 』の「Easy Virtual Network の概要」モジュール
Easy Virtual Network の設定	『 <i>Easy Virtual Network Configuration Guide</i> 』の「Easy Virtual Network の設定」モジュール
Easy Virtual Network の管理とトラブルシューティング	『 <i>Easy Virtual Network Configuration Guide</i> 』の「Easy Virtual Network の管理とトラブルシューティング」モジュール

MIB

MIB	MIB のリンク
<p>VRF 情報を提供する MIB は引き続き Easy Virtual Network で動作します。VRF から独立した MIB は、システムのすべての VRF に関する次の情報をレポートします。</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>★枠で囲まれた Technical Assistance の場合★右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする場合、Cisco.com のログイン ID およびパスワードが必要です。</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

Easy Virtual Network 共有サービスの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 7: *Easy Virtual Network* 共有サービスの機能情報

機能名	リリース	機能情報
EVN ルート レプリケーション	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	<p>このモジュールでは、EVN 環境でルート レプリケーションと再配布を使用してサービスを共有する方法について説明します。</p> <p>この機能によって、次のコマンドが変更されました。 redistribute (IP)</p> <p>この機能によって、次のコマンドが導入されました。 route-replicate (VRF アドレスファミリ)</p>

