



Easy Virtual Network Configuration Guide, Cisco IOS Release 15SY

First Published: October 15, 2012

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2014 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Overview of Easy Virtual Network 1

Finding Feature Information 1

Prerequisites for Configuring EVN 2

Restrictions for EVN 2

Information About EVN 3

Benefits of EVN 3

Virtual Network Tags Provide Path Isolation 4

Virtual Network Tag 5

vnet Global 6

Edge Interfaces and EVN Trunk Interfaces 6

Identifying Trunk Interfaces in Display Output 7

Single IP Address on Trunk Interfaces 8

Relationship Between VRFs Defined and VRFs Running on a Trunk Interface 8

VRF Awareness 9

Routing Protocols Supported by EVN 9

Packet Flow in a Virtual Network 10

Command Inheritance on EVN Trunk Interfaces 11

Overriding Command Inheritance Virtual Network Interface Mode 11

Example: Overriding Command Inheritance 12

Example: Enabling an Attribute to vnet Global Only 12

Removing Overrides and Restoring Values Inherited from EVN Trunk 12

Determining if No Form of Command Appears in Configuration File 13

EXEC Commands Routing Context 14

EVN Compatibility with VRF-Lite 14

Multiaddress Family VRF Structure 15

QoS Functionality with EVN 15

Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface 15

Additional References	19
Feature Information for Overview of Easy Virtual Network	21

CHAPTER 2

Configuring Easy Virtual Network	23
Finding Feature Information	23
Prerequisites for Configuring EVN	23
How to Configure EVN	24
Configuring an Easy Virtual Network Trunk Interface	24
Enabling a Subset of VRFs over a Trunk Interface	28
Configuring an EVN Edge Interface	30
What to Do Next	32
Verifying EVN Configurations	32
Configuration Examples for Configuring EVN	33
Example: Virtual Networks Using OSPF with network Commands	33
Example: Virtual Networks Using OSPF with ip ospf vnet area Command	34
Example: Command Inheritance and Virtual Network Interface Mode Override in an EIGRP Environment	34
Example: Command Inheritance and Virtual Network Interface Mode Override in a Multicast Environment	37
Example: EVN Using IP Multicast	38
Additional References	39
Feature Information for Configuring Easy Virtual Network	40

CHAPTER 3

Easy Virtual Network Management and Troubleshooting	43
Finding Feature Information	43
Prerequisites for EVN Management and Troubleshooting	43
Information About EVN Management and Troubleshooting	44
Routing Context for EXEC Mode Reduces Repetitive VRF Specification	44
Output of traceroute Command Indicates VRF Name and VRF Tag	44
Debug Output Filtering Per VRF	45
CISCO-VRF-MIB	45
How to Manage and Troubleshoot EVN	45
Setting the Routing Context for EXEC Mode to a Specific VRF	45
Enabling Debug Output for VRFs	47
Setting SNMP v2c Context for Virtual Networks	47

Setting SNMP v3 Context for Virtual Networks	48
Additional References	50
Feature Information for EVN Management and Troubleshooting	51

CHAPTER 4

Configuring Easy Virtual Network Shared Services	53
Finding Feature Information	53
Prerequisites for Virtual IP Network Shared Services	53
Restrictions for Virtual IP Network Shared Services	54
Information About Easy Virtual Network Shared Services	54
Shared Services in an Easy Virtual Network	54
Easy Virtual Network Shared Services Easier than VRF-Lite	54
Route Replication Process in Easy Virtual Network	54
Where to Implement Route Replication	56
Route Replication Behavior for Easy Virtual Network	56
Route Preference Rules After Route Replication in Easy Virtual Network	56
How to Share Services Using Easy Virtual Network	57
Configuring Route Replication to Share Services in Easy Virtual Network	57
Example	63
What to Do Next	63
Configuring Redistribution to Share Services in Easy Virtual Network	64
Configuration Example for Easy Virtual Network Shared Services	67
Example: Easy Virtual Network Route Replication and Route Redistribution in a Multicast Environment	67
Additional References	72
Feature Information for Easy Virtual Network Shared Services	73



Overview of Easy Virtual Network

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization of two or more Layer-3 networks. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other.

EVN builds on the existing IP-based virtualization mechanism known as VRF-Lite. EVN provides enhancements in path isolation, simplified configuration and management, and improved shared service support. EVN is backward compatible with VRF-Lite to enable seamless network migration from VRF-Lite to EVN.

EVN supports IPv4, static routes, Open Shortest Path First version 2 (OSPFv2), and Enhanced Interior Gateway Routing Protocol (EIGRP) for unicast routing, and Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) for IPv4 Multicast routing. EVN also supports Cisco Express Forwarding (CEF) and Simple Network Management Protocol (SNMP).

- [Finding Feature Information, page 1](#)
- [Prerequisites for Configuring EVN, page 2](#)
- [Restrictions for EVN, page 2](#)
- [Information About EVN, page 3](#)
- [Additional References, page 19](#)
- [Feature Information for Overview of Easy Virtual Network, page 21](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring EVN

- Implementing EVN in a network requires a single IP infrastructure that you want to virtualize into two or more logical networks or L3VPNs. EVN provides path isolation for the traffic on the different virtual networks.
- You must have a functioning campus design in place before adding virtualization to a network.
- You should understand virtual routing and forwarding (VRF) instances and how they are used to maintain traffic separation across the network.

Restrictions for EVN

- An EVN trunk is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels.
- There are additional platform and line-card restrictions for an EVN trunk. Check Cisco Feature Navigator, www.cisco.com/go/cfn for supported platforms and line cards.
- A single IP infrastructure can be virtualized to provide up to 32 virtual networks end-to-end.
- If an EVN trunk is configured on an interface, you cannot configure VRF-Lite on the same interface.
- OSPFv3 is not supported; OSPFv2 is supported.
- The following are not supported by EVN:
 - IS-IS
 - RIP
 - Route replication is not supported with BGP
 - Certain SNMP set operations
- The following are not supported on an EVN trunk:
 - Access control lists (ACLs)
 - BGP interface commands are not inherited
 - IPv6, except on vnet global
 - Network address translation (NAT)
 - NetFlow
 - Web Cache Communication Protocol (WCCP)

Information About EVN

Benefits of EVN

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization over Layer-3 networks. Network virtualization can be used to secure a network and to reduce network expenses by utilizing the same network infrastructure for multiple virtual networks. You can leverage the same physical infrastructure multiple times by supporting multiple groups, each with their own logical network and unique routing and forwarding tables.

Prior to network virtualization, path isolation can be achieved by:

- Separating paths using dedicated routers which is more expensive than virtual networks.
- Using access control lists (ACLs), but ACLs do not support unique routing and forwarding tables, can be expensive to maintain, and more prone to error than virtual networks.

EVN provides the following benefits:

- Reduced capital expenditures by not having to maintain separate physical infrastructures to keep traffic isolated. One IP network has two or more virtual networks with traffic path isolation thereby saving the expense of additional hardware.
- Increased business flexibility, due to the ease of network integration for mergers, acquisitions, and business partners.
- Reduced network complexity due to a decrease in the infrastructure requirements for maintaining traffic separation through the core of the network.
- Build on the existing mechanism known as Multi-VRF (VRF-Lite). EVN is compatible with VRF-Lite. See the EVN Compatibility with VRF-Lite section. EVN is recommended over VRF-Lite because EVN provides enhancements in path isolation, simplified configuration and management, and improved shared service support.

In addition to maintaining traffic separation between business units within a company, there are other scenarios in which path isolation is beneficial, including the following:

- Guest access to the Internet—Restricting a guest's network access to the Internet, using a predetermined data path through the customer's network, and being able to define a unique default route for guest traffic.
- Network Admission Control (NAC) isolation—Isolating the traffic sourced from a noncompliant desktop.
- Partner access—Restricting partners and contractors to access a network's shared services, such as the Internet, e-mail, DNS, DHCP, or an application server.
- Application and device isolation—Securing services and devices by "forcing" traffic to a centralized firewall where the traffic is subject to inspection.
- Outsourcing services—Separating data traffic of various clients from each other.
- Scalable network—Restricting a portion of the network to traffic that requires a very strict service level, which can lower costs by providing those requirements only where needed.

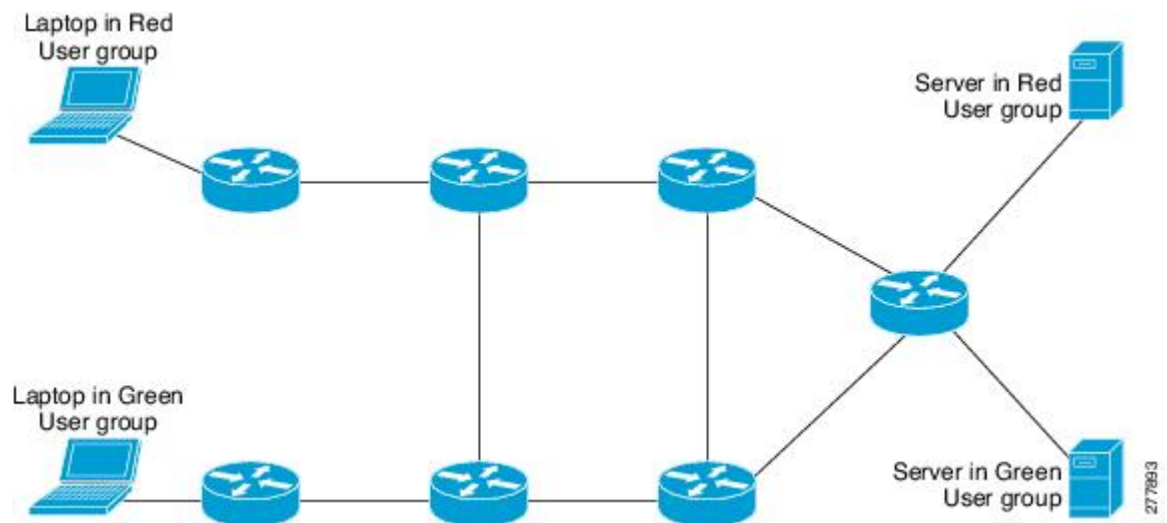
- Subsidiaries/mergers/acquisitions—Consolidating companies or networks in stages, while enabling them to share services, when required.
- Enterprise acting as a service provider—Requiring a separate network under a single authority for autonomous groups. An example is an airport authority supporting a virtual network per airline.

Virtual Network Tags Provide Path Isolation

It is not uncommon to have different user groups running on the same IP infrastructure. Various business reasons require traffic isolation between different groups. The figure below shows two user groups, Red and Green, running on the same network. Prior to network virtualization, there is no separation of traffic between the two groups. Users in the Red user group can access the server in the Green user group, and vice versa.

Without network virtualization, path isolation can be achieved by using access control, which is expensive to maintain, prone to error and does not support unique routing and forwarding tables per network.

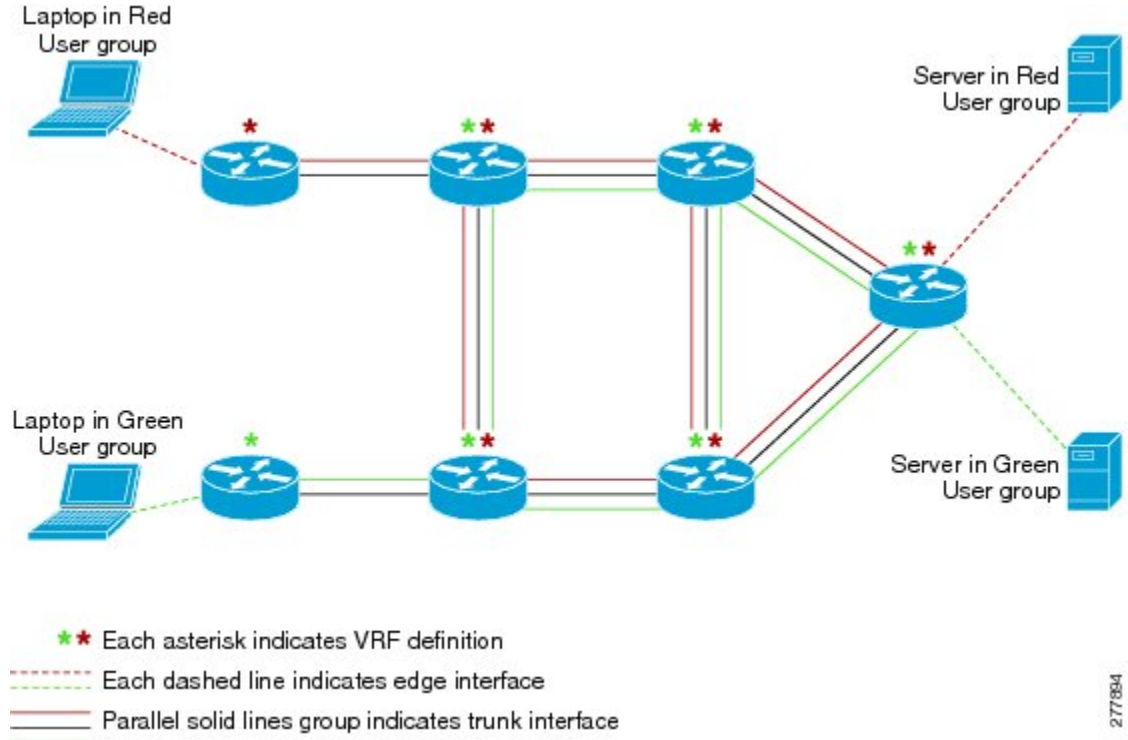
Figure 1: Network without Virtualization



Virtual networks provide a coarse-grained segmentation of different user groups on one physical network. By configuring virtual networks, you can virtualize a single IP infrastructure to provide a number of virtual

networks end to end. In the figure below, a single IP infrastructure is virtualized into two VPNs by creating two VRFs, Red and Green.

Figure 2: Network with Virtualization



277884

In addition to utilizing VRFs to provide device-level separation, each virtual network has path isolation from the other. Path isolation is achieved by tagging the traffic so it carries the same tag value throughout the same virtual network. Each network device along the path uses the tags to provide separation among different VRFs. A single tag number ties VRF red, for example, on one router to VRF red on another router.

Virtual Network Tag

Each VPN and associated EVN has a tag value that you assign during configuration. The tag value is global, meaning that on each router, the same EVN must be assigned the same numerical tag value. Tag values range from 2 to 4094.



Note

When configuring EVN on a Cisco Catalyst 6500 Family networking device, we recommend you assign a vnet tag in the range 2 to 1000. Beginning with Cisco IOS Release 15.1(1)SY, on the Sup2T platform of the Cisco Catalyst 6000 product lines, if the **vlan internal allocation policy descending** command is configured, the **vnet tag** range is from 2 to 3900.

An EVN is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels. To allow for backward compatibility with the VRF-Lite solution, the vLAN ID field in the 802.1q frame is used to carry the virtual network tag.

Traffic that carries a virtual network tag is called tagged traffic. Traffic that does not carry a virtual network tag is called untagged traffic.

Tags are illustrated in the following configuration with two VRFs, red and green:

```
! 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
  !
```

A virtual network is defined as a VRF instance with a virtual network tag assigned.

vnet Global

A predefined EVN known as “vnet global” is on the device. It refers to the global routing context and it corresponds to the default RIB. In figure 2 and figure 3, vnet global is represented by a black line connecting routers. The vnet global carries untagged traffic. By default, interfaces belong to the vnet global. Furthermore, vnet global is always running on trunk interfaces. The vnet global is also known as the default routing table.



Note

IPv6 traffic is supported in vnet global only.

Edge Interfaces and EVN Trunk Interfaces

User devices are connected to a Layer 2 switch port, which is assigned to a VLAN. A VLAN can be thought of as a Layer 2 VPN. Customers will group all of the devices that need to be supported in a common Layer 3 VPN in a single VLAN. The point where data traffic is handed off between a VLAN and VRF is called an edge interface.

- An edge interface connects a user device to the EVN and in effect defines the boundary of the EVN. Edge interfaces connect end devices such as hosts and servers that are not VRF-aware. Traffic carried over the edge interface is untagged. The edge interface classifies which EVN the received traffic belongs to. Each edge interface is configured to belong to only one EVN.
- An EVN trunk interface connects VRF-aware routers together and provides the core with a means to transport traffic for multiple EVNs. Trunk interfaces carry tagged traffic. The tag is used to de-multiplex the packet into the corresponding EVN. A trunk interface has one subinterface for each EVN. The **vnet trunk** command is used to define an interface as an EVN trunk interface.

An EVN interface uses two types of interfaces: edge interfaces and trunk interfaces. An interface can be an edge or trunk interface, but not both. Figure 3 illustrates Routers A and D, which have edge interfaces that belong to VRF Red. Routers D and E have edge interfaces that belong to VRF Green.

You can display this hidden interface with the **show derived-config** command and see that all of the commands entered on Fast Ethernet 0/0/0 have been inherited by Fast Ethernet 0/0/0.3:

```
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
```

Single IP Address on Trunk Interfaces

A trunk interface can carry traffic for multiple EVNs. To simplify the configuration process, all the subinterfaces and associated EVNs have the same IP address assigned. In other words, a trunk interface is identified by the same IP address in different EVN contexts. This is because each EVN has a unique routing and forwarding table, thereby enabling support for overlapping IP addresses across multiple EVNs.

Relationship Between VRFs Defined and VRFs Running on a Trunk Interface

By default, the trunk interfaces on a router will carry traffic for all VRFs defined by the **vrf definition** command. For example, in the following configuration, every VRF defined on the router is included on the interface:

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

However, you might want to enable only a subset of VRFs over a certain trunk interface for traffic separation purposes. This is achieved by creating a VRF list, which is referenced in the **vnet trunk** command. When a trunk interface is enabled with a VRF list, only VRFs on the list are enabled on the interface. The exception is that **vnet global** is always enabled on the trunk interface.

In the following example, only the two specified VRFs on the list (red and green) are enabled on the interface:

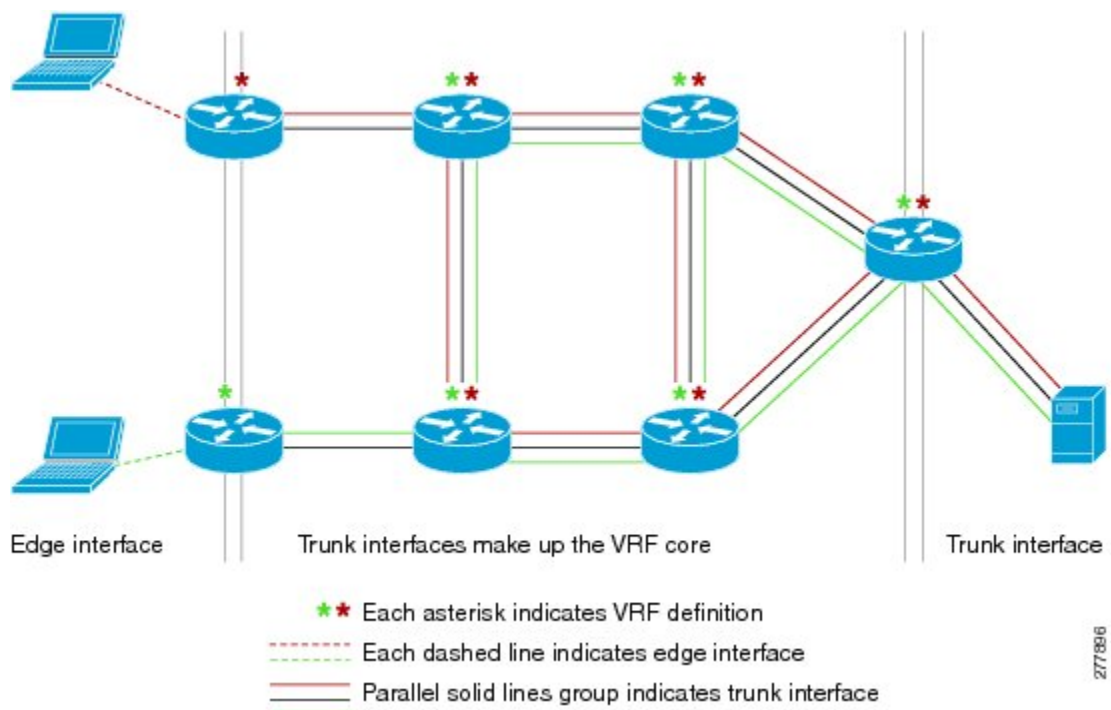
```
vrf list mylist
member red
member green
!
interface FastEthernet 1/0/0
vnet trunk list mylist
ip address 10.1.1.1 255.255.255.0
```

VRF Awareness

A device connected to a virtual network may not understand virtual network tags and can send and receive only untagged traffic. Such a device is referred to as VRF unaware. For example, a laptop computer is usually VRF unaware.

By contrast, a device that can send and receive tagged traffic and therefore takes the tag value into account when processing such traffic is known as VRF aware. For example, a VRF-aware server shared among different EVNs could use the virtual network tag to distinguish requests received and send responses. A VRF-aware device is connected to the EVN using a trunk interface, as shown in figure 4.

Figure 4: VRF Aware Server



The term “VRF aware” can also be used to describe a software component running on the router. A software component is VRF aware if it can operate on different EVNs. For example, ping is VRF aware because it allows you to choose which EVN to send the ping packet over.

Routing Protocols Supported by EVN

Each EVN runs a separate instance of a routing protocol. This allows each EVN to fine-tune its routing separately and also limits fate sharing. Different virtual networks may run different routing protocols concurrently.

EVN supports static routes, OSPFv2, and EIGRP for unicast routing, and PIM, MSDP, and IGMP for multicast routing.

Packet Flow in a Virtual Network

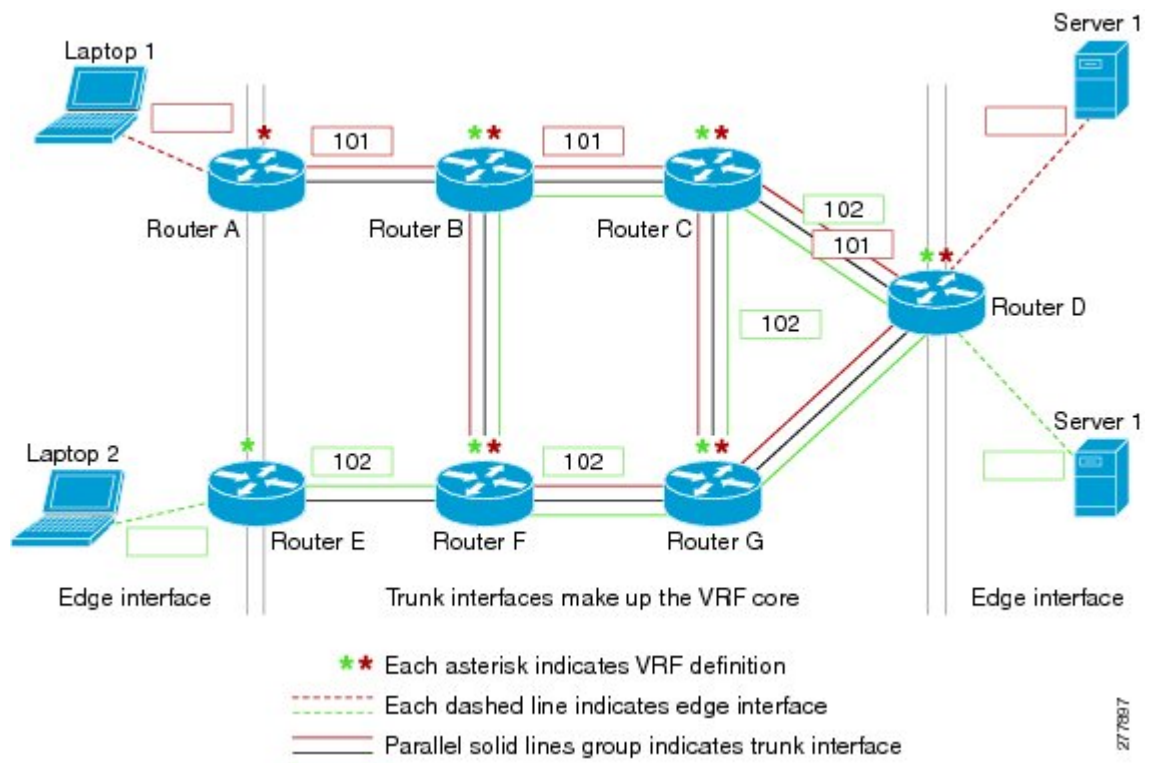
Packets enter an EVN through an edge interface, traverse multiple trunk interfaces, and exit the virtual network through another edge interface. At the ingress edge interface, packets are mapped from a VLAN into a particular EVN. Once the packet is mapped to an EVN, it is tagged with the associated virtual network tag. The virtual network tag allows the trunk interface to carry packets for multiple EVNs. The packets remain tagged until they exit the EVN through the egress edge interface.

On the edge interface, the EVN associated with the interface is used for route lookup. On the trunk interface, the virtual network tag carried in the packet is used to locate the corresponding EVN for routing the packets.

If the egress interface is an edge interface, the packet is forwarded untagged. However, if the egress interface is a trunk interface, the packet is forwarded with the tag of the ingress EVN.

The figure below illustrates how traffic from two VRFs, red and green, can coexist on the same IP infrastructure, using the tags 101 and 102.

Figure 5: Packet Flow in a Virtual Network



The packet flow from Laptop 1 to Server 1 in VRF red occurs as follows:

- 1 Laptop 1 send an untagged packet to Server 1.
- 2 Router A receives the packet over an edge interface, which is associated with VRF red.
 - 1 Router A does route lookup in VRF red and sees that the next hop is Router B through a trunk interface.
 - 2 Router A encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.

- 3 Router B receives the packet over a trunk interface. Seeing virtual network tag 101, Router B identifies that the packet belongs to VRF red.
 - 1 Router B does route lookup in VRF red and sees that the next hop is Router C through a trunk interface.
 - 2 Router B encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
- 4 Router C receives the packet over a trunk interface. Using virtual network tag 101, Router C identifies that the packet belongs to VRF red.
 - 1 Router C does route lookup in VRF red and sees that the next hop is Router D through a trunk interface.
 - 2 Router C encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
- 5 Router D receives the packet over a trunk interface. Using virtual network tag 101, Router D identifies that the packet belongs to VRF red.
 - 1 Router D does route lookup in VRF red and sees that the next hop is through an edge interface.
 - 2 Router D sends the untagged packet over the edge interface to Server 1.
- 6 Server 1 receives the untagged packet originated from Laptop 1.

Command Inheritance on EVN Trunk Interfaces

One of the benefits of EVN is the ability to easily configure multiple EVNs on a common trunk interface without the need to configure each interface associated with an EVN individually. An EVN trunk interface takes advantage of the fact that the configuration requirements for different EVNs will be similar over a single trunk interface. When specific commands are configured on the trunk interface, they define default values that are inherited by all EVNs running over the same interface, including **vnet global**. If you feel that the settings are acceptable for all of the EVNs sharing an interface, then no individual configuration is necessary.

For example, the OSPF hello interval can be set for all EVNs over the trunk interface with one line of configuration, as follows:

```
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
```

The list of commands configured on the trunk interface whose values are inherited by all EVNs running on the same interface is provided in the table in "Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface" section.

For more examples of command inheritance, see the configuration examples in the *Configuring Easy Virtual Networks* module.

Overriding Command Inheritance Virtual Network Interface Mode

You might want some EVNs on the same trunk interface to have different configurations. An alternative to command inheritance is to selectively override inherited values by using specific commands in virtual network interface mode for individual EVNs. In this mode, the command's settings override the Cisco default value or the value you set in interface configuration mode.

In interface configuration mode, entering the **vnet name** command causes the system to enter virtual network interface mode. The system prompt for this mode is Router(config-if-vnet)#.

The list of commands whose inherited values can be overridden is provided in the table in the “Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface” section in this module.

Example: Overriding Command Inheritance

In the following example, the OSPF cost of 30 for VRF blue overrides the OSPF cost of 20 for the other VRFs on the interface:

```
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
```

The **show derived** command indicates the subinterface changed to a cost of 30:

```
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)#
```

Example: Enabling an Attribute to vnet Global Only

Similarly, you might want to enable an attribute to vnet global only. To do so, use the **vnet global** interface submode, as follows:

```
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
```

In this example, a user wants an EIGRP interface attribute set for all EVNs except vnet global. All EVNs inherit a hold time of 20 seconds, except vnet global, which overrides 20 with a hold time of 40 seconds.

```
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
```

Removing Overrides and Restoring Values Inherited from EVN Trunk

The **no** and **default** keywords result in different outcomes, depending on whether they are used for a trunk interface or in virtual network interface mode. This section describes the different outcomes.

- When the **no** or **default** keyword is entered before a command on a trunk interface, the trunk is restored to the system's default value for that command. (This is standard behavior resulting for the **no** or **default** keyword).
- When the **default** keyword is entered before a command in virtual network interface mode, the override value is removed and the value that is inherited from the trunk is restored. The override value for the specific EVN is no longer in effect.

In the following example, the trunk interface is configured with an OSPF cost of 20, but VRF blue overrides that value with an OSPF cost of 30:

```
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
```

When the following commands are entered, the OSPF cost value is restored to 20, which is the cost inherited from the trunk interface. (Note that 20 is not the default value of the **ip ospf cost** command.)

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

The **default** keyword entered before a command in virtual network interface mode restores the default state, but the **no** keyword does not always do that. In the following example, **no ip dampening-change eigrp 1** disables dampening change.

```
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.
 !
```

Determining if No Form of Command Appears in Configuration File

If a command is the type of command that switches a feature on or off, the **no** form of the command will appear in the configuration file when configured. That is, nonvolatile generation (NVGEN) overrides the setting from the EVN trunk, as shown in the following example:

```
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
```

If a command takes an argument in its syntax, such as **ip ospf cost cost**, the **no** form of the command will remove the configuration, but does not appear in the configuration file. That is, it will not be NVGEN'ed because the user could enter **ip ospf cost default-value** to override the inherited value in a more direct way.

EXEC Commands Routing Context

There may be occasions when you want to issue several EXEC commands to apply to a single EVN. In order to reduce the repetitive entering of VRF names for multiple EXEC commands, the **routing-context vrf** command allows you to set the VRF context of EXEC commands once, and then proceed using EXEC commands.

The table below shows four EXEC commands without routing context and in routing context. Note that in the left column, each EXEC command must identify the VRF. In the right column, the VRF content is identified once and the prompt changes to reflect that VRF; there is no need to identify the VRF in each command.

Table 1: EXEC Commands Routing Context

EXEC Commands Without Routing Context	EXEC Commands Routing Context
—	Router# routing-context vrf red Router%red#
Router# show ip route vrf red [Routing table output for VRF red]	Router%red# show ip route [Routing table output for VRF red]
Router# ping vrf red 10.1.1.1 [Ping result using VRF red]	Router%red# ping 10.1.1.1 [Ping result using VRF red]
Router# telnet 10.1.1.1 /vrf red [Telnet to 10.1.1.1 in VRF red]	Router%red# telnet 10.1.1.1 [Telnet to 10.1.1.1 in VRF red]
Router# traceroute vrf red 10.1.1.1 [Traceroute output in VRF red]	Router%red# traceroute 10.1.1.1 [Traceroute output in VRF red]

EVN Compatibility with VRF-Lite

EVN is wire compatible with VRF-Lite. In other words, on the outside, 802.1q, SNMP MIBs, and all the EVN infrastructure will look exactly the same as VRF-Lite.

In the figure below, both routers have VRFs defined. The router on the left uses VRF-Lite, and the router on the right uses an EVN trunk with tags. The two configurations follow the figure.



VRF-Lite Subinterface Configuration EVN Trunk Configuration

```

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
 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

```

```

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

```

Multiaddress Family VRF Structure

Prior to Cisco IOS Releases 12.2(33)SB and 15.0(1)M, the CLI for a VRF applied to only one address family at a time. For example, the **ip vrf blue** command applies only to the IPv4 address family.

In Cisco IOS Releases 12.2(33)SB and 15.0(1)M, the CLI for a VRF applies to multiple address families under the same VRF. This is known as multiprotocol VRF. For example, the **vrf definition blue** command applies to IPv4 and IPv6 VPNs at the same time, but the routing tables for the two protocols are still different.



Note

In Cisco IOS XE Release 3.2S, virtual networks do not support IPv6 except in **vnet global**.

QoS Functionality with EVN

Quality of Service (QoS) configurations are applied to the main physical interface on an EVN trunk. The QoS policy affects all traffic that flows out the physical interface in all the VRFs at the same time. In other words, QoS and network virtualization are mutually independent. For example, traffic marked with the DSCP value specified for voice will be put into the voice queue if the packet is from the red VRF, blue VRF, or green VRF. The traffic for all the VRFs will be queued together.

Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface

As explained in the "Command Inheritance on EVN Trunk Interfaces" section, there are interface commands that are defined once for a trunk interface, and the value is inherited by each EVN sharing the interface. These commands are sometimes referred to as trunk commands.

A subset of the trunk commands are commands whose values can be overridden by specifying the command in virtual network interface mode. This is explained in the "Overriding Command Inheritance Virtual Network Interface Mode" section.

The table below lists interface commands and indicates whether the values are inherited by the EVNs on the interface and whether the commands can be overridden for a specific EVN.

Table 2: Interface Command Values Inherited or Overridden by a Virtual Network on an Interface

	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
IP Commands		
ip accounting	Yes	No
ip address	Yes	No
ip broadcast-address	Yes	No
ip directed broadcast	Yes	No
ip information-reply	Yes	No
ip irdp	Yes	No
ip load-sharing	Yes	No
ip mask-reply	Yes	No
ip mtu	Yes	No
ip proxy-arp	Yes	No
ip redirects	Yes	No
ip unnumbered	Yes	No
ip unreachable	Yes	No
EIGRP Commands		
ip authentication key-chain eigrp	Yes	Yes
ip authentication mode eigrp	Yes	Yes
ip bandwidth-percent eigrp	Yes	Yes
ip dampening-change eigrp	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip dampening-interval eigrp	Yes	Yes
ip hello-interval eigrp	Yes	Yes
ip hold-time eigrp	Yes	Yes
ip next-hop-self eigrp	Yes	Yes
ip split-horizon eigrp	Yes	Yes
ip summary-address eigrp	Yes	Yes
Commands that Affect how EIGRP Determines Cost for an Interface		
bandwidth (interface)	Yes	Yes
delay (interface)	Yes	Yes
OSPF Commands		
ip ospf <i>process-id</i> area	No	Yes
ip ospf authentication	Yes	Yes
ip ospf authentication-key	Yes	Yes
ip ospf bfd	Yes	Yes
ip ospf cost	Yes	Yes
ip ospf database-filter	Yes	Yes
ip ospf dead-interval	Yes	Yes
ip ospf demand-circuit	Yes	Yes
ip ospf flood-reduction	Yes	Yes
ip ospf hello-interval	Yes	Yes
ip ospf lls	Yes	Yes
ip ospf message-digest-key	Yes	Yes
ip ospf mtu-ignore	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip ospf network	Yes	Yes
ip ospf priority	Yes	Yes
ip ospf resync-timeout	Yes	Yes
ip ospf shutdown	Yes	Yes
ip ospf transmit-delay	Yes	Yes
ip ospf transmit-interval	Yes	Yes
ip ospf ttl-security	Yes	Yes
ip ospf vnet area	No	No
IP Multicast Commands		
ip igmp access-group	Yes	Yes
ip igmp explicit-tracking	Yes	Yes
ip igmp helper-address	Yes	Yes
ip igmp immediate-leave	Yes	Yes
ip igmp last-member-query-count	Yes	Yes
ip igmp last-member-query-interval	Yes	Yes
ip igmp limit	Yes	Yes
ip igmp mroute-proxy	Yes	Yes
ip igmp proxy-service	Yes	Yes
ip igmp querier-timeout	Yes	Yes
ip igmp query-interval	Yes	Yes
ip igmp query-max-response-time	Yes	Yes
ip igmp tcn	Yes	Yes
ip igmp unidirectional-link	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip igmp v3lite	Yes	Yes
ip igmp version	Yes	Yes
ip multicast boundary	Yes	Yes
ip pim bidir-neighbor-filter	Yes	Yes
ip pim bsr-border	Yes	Yes
ip pim dense-mode	Yes	Yes
ip pim dr-priority	Yes	Yes
ip pim nbma-mode	Yes	Yes
ip pim neighbor-filter	Yes	Yes
ip pim passive	Yes	Yes
ip pim query-interval	Yes	Yes
ip pim sparse-dense-mode	Yes	Yes
ip pim sparse-mode	Yes	Yes
ip pim state-refresh	Yes	Yes
Multicast Forwarding Information Base (MFIB) Commands		
ip mfib cef	Yes	Yes
ip mfib forwarding	Yes	Yes

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Easy Virtual Network commands	Easy Virtual Network Command Reference
Configuring Easy Virtual Network	“Configuring Easy Virtual Network” module in the <i>Easy Virtual Network Configuration Guide</i>
Configuring Easy Virtual Network shared services and route replication	“Configuring Easy Virtual Network Shared Services” module in the <i>Easy Virtual Network Configuration Guide</i>
Easy Virtual Network management and troubleshooting	“Easy Virtual Network Management and Troubleshooting” module in the <i>Easy Virtual Network Configuration Guide</i>

MIBs

MIB	MIBs Link
<p>Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system.</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

Feature Information for Overview of Easy Virtual Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for Overview of Easy Virtual Network

Feature Name	Releases	Feature Information
EVN VNET Trunk	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 is an IP-based virtualization technology that provides end-to-end virtualization of the network. You can use a single IP infrastructure to provide separate virtual networks with isolated traffic paths.



CHAPTER 2

Configuring Easy Virtual Network

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end network virtualization. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other. Configure Easy Virtual Network to configure two or more virtual IP networks.

- [Finding Feature Information, page 23](#)
- [Prerequisites for Configuring EVN, page 23](#)
- [How to Configure EVN, page 24](#)
- [Configuration Examples for Configuring EVN, page 33](#)
- [Additional References, page 39](#)
- [Feature Information for Configuring Easy Virtual Network, page 40](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring EVN

- Implementing EVN in a network requires a single IP infrastructure that you use to create two or more virtual networks. You want path isolation for traffic on the different virtual networks.
- You should understand the concepts in the “Overview of Easy Virtual Network” module.
- We recommend that you draw your network topology, indicating the interfaces on each router that belong to the EVNs. The diagram facilitates tracking the interfaces you are configuring as edge interfaces and the interfaces you are configuring as trunk interfaces.

How to Configure EVN

Configuring an Easy Virtual Network Trunk Interface

Perform this task to configure an EVN trunk interface, which connects routers to provide the core to transport traffic for multiple virtual networks. Traffic carried over a trunk interface is tagged. This task illustrates how to configure a trunk interface with a base virtual routing and forwarding (VRF) and two named VRFs: VRF red and VRF blue.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit-address-family**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *number*
11. **description** *string*
12. **address-family ipv4**
13. **exit-address-family**
14. **exit**
15. **interface** *type number*
16. **ip address** *ip-address mask*
17. **vnet trunk** [**list** *vrf-list-name*]
18. **vnet name** *vrf-name*
19. **exit-if-vnet**
20. **no shutdown**
21. **exit**
22. **router ospf** *process-id*
23. **network** *ip-address wildcard area area-id*
24. **exit**
25. **router ospf** *process-id vrf vrf-name*
26. **network** *ip-address wildcard area area-id*
27. **exit**
28. **router ospf** *process-id vrf vrf-name*
29. **network** *ip-address wildcard area area-id*
30. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition red	Configures a VRF routing table instance and enters VRF configuration mode.
Step 4	vnet tag <i>number</i> Example: Router(config-vrf)# vnet tag 100	Specifies the global numeric tag for the VRF. <ul style="list-style-type: none"> • The same tag number must be configured for the same virtual network on each edge and trunk interface. • When configuring EVN on a Cisco Catalyst 6500 family networking device, we recommend you assign a vnet tag number in the range 2 to 1000.
Step 5	description <i>string</i> Example: Router(config-vrf)# description guest access	(Optional) Describes a VRF to help a network administrator review the configuration files.
Step 6	address-family ipv4 Example: Router(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IP version 4 address prefixes.
Step 7	exit-address-family Example: Router(config-vrf-af)# exit-address-family	Exits address family configuration mode.
Step 8	exit Example: Router(config-vrf)# exit	Exits to global configuration mode.

	Command or Action	Purpose
Step 9	vrf definition <i>vrf-name</i> Example: <pre>Router(config)# vrf definition blue</pre>	Configures a VRF routing table instance and enters VRF configuration mode.
Step 10	vnet tag <i>number</i> Example: <pre>Router(config-vrf)# vnet tag 200</pre>	Specifies the global numeric tag for the VRF. <ul style="list-style-type: none"> • The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 11	description <i>string</i> Example: <pre>Router(config-vrf) description Finance</pre>	(Optional) Describes a VRF to help a network administrator review configuration files.
Step 12	address-family <i>ipv4</i> Example: <pre>Router(config-vrf) address-family ipv4</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 13	exit-address-family Example: <pre>Router(config-vrf-af) exit-address-family</pre>	Exits address family configuration mode.
Step 14	exit Example: <pre>Router(config-vrf)# exit</pre>	Exits to global configuration mode.
Step 15	interface <i>type number</i> Example: <pre>Router(config)# interface gigabitethernet 1/1/1</pre>	Configures an interface type and enters interface configuration mode.
Step 16	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 10.1.1.1 255.255.255.0</pre>	Sets a primary IP address for the interface.
Step 17	vnet trunk [<i>list vrf-list-name</i>] 	Defines a trunk interface.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-if)# vnet trunk</pre>	<ul style="list-style-type: none"> By default, all VRFs defined with the vrf definition command run on all trunk interfaces on the router. Therefore, VRF red and VRF blue are now running on this interface. Use the list vrf-list-name command elements to restrict VRFs running on a trunk interface.
Step 18	<p>vnet name vrf-name</p> <p>Example:</p> <pre>Router(config-if)# vnet name red</pre>	<p>(Optional) Enters virtual network interface mode to configure features that apply to a specified VRF to override global VRF values.</p> <ul style="list-style-type: none"> This step is not necessary if the global settings are acceptable for all of the VRFs on the interface. After this step, you configure one or more eligible commands, such as ip ospf cost. (Not shown in this task.) For the list of commands that are used to override global VRF values, see Overview of Easy Virtual Network module, Table 2.
Step 19	<p>exit-if-vnet</p> <p>Example:</p> <pre>Router(config-if-vnet) exit-if-vnet</pre>	Exits VRF interface configuration mode and enters interface configuration mode.
Step 20	<p>no shutdown</p> <p>Example:</p> <pre>Router(config-if) no shutdown</pre>	Restarts an interface.
Step 21	<p>exit</p> <p>Example:</p> <pre>Router(config-if) exit</pre>	Exits to global configuration mode.
Step 22	<p>router ospf process-id</p> <p>Example:</p> <pre>Router(config)# router ospf 1</pre>	<p>Configures an Open Shortest Path First (OSPF) routing process and associates it with a VRF.</p> <ul style="list-style-type: none"> This OSPF instance has no VRF, so it is vnet global.
Step 23	<p>network ip-address wildcard area area-id</p> <p>Example:</p> <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 0</pre>	Defines the interfaces and associated area IDs on which OSPF runs.

	Command or Action	Purpose
Step 24	exit Example: <pre>Router(config-router) exit</pre>	Exits to global configuration mode.
Step 25	router ospf <i>process-id</i> vrf <i>vrf-name</i> Example: <pre>Router(config)# router ospf 2 vrf red</pre>	Configures an OSPF routing process and associates it with a VRF. <ul style="list-style-type: none"> • Specifies a different <i>process-id</i> for each VRF because they each need their own OSPF instance.
Step 26	network <i>ip-address wildcard</i> area <i>area-id</i> Example: <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 0</pre>	Defines the interfaces and associated area IDs on which OSPF runs and the area ID for those interfaces.
Step 27	exit Example: <pre>Router(config-router) exit</pre>	Exits to global configuration mode.
Step 28	router ospf <i>process-id</i> vrf <i>vrf-name</i> Example: <pre>Router(config)# router ospf 3 vrf blue</pre>	Configures an OSPF routing process and associates it with a VRF. <ul style="list-style-type: none"> • Specifies a different <i>process-id</i> for each VRF because they each need their own OSPF instance.
Step 29	network <i>ip-address wildcard</i> area <i>area-id</i> Example: <pre>Router(config-router) network 10.0.0.0 255.255.255.0 area 2</pre>	Defines the interfaces and associated area IDs on which OSPF runs and the area ID for those interfaces.
Step 30	end Example: <pre>Router(config-vrf) end</pre>	Ends the configuration session and returns to privileged EXEC mode.

Enabling a Subset of VRFs over a Trunk Interface

The prior task, “Configuring an Easy Virtual Network Trunk Interface,” shows how to configure a trunk interface with two VRFs. By default, the trunk interfaces on a router can carry traffic for each VRF defined by the **vrf definition** command. However, you might want to enable only a subset of VRFs over a trunk

interface, which is done by creating a VRF list. A maximum of 32 VRF lists can exist on a router. Perform the following task to create a VRF list. This task presumes that the VRF has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf list** *vrf-list-name*
4. **member** *vrf-name*
5. Repeat Step 4 to add other VRFs to the list.
6. **exit-vrf-list**
7. **interface** *type number*
8. **vnet trunk list** *vrf-list-name*
9. **ip address** *ip-address mask*
10. **end**
11. **show vrf list** [*vrf-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	vrf list <i>vrf-list-name</i> Example: Router(config)# vrf list External	Defines a list of VRFs and enters VRF list configuration mode. <ul style="list-style-type: none"> • The <i>vrf-list-name</i> argument may contain up to 32 characters. Quotation marks, spaces, and * are not allowed.
Step 4	member <i>vrf-name</i> Example: Router(config-vrf-list)# member blue	Specifies an existing VRF as a member of a VRF list. <ul style="list-style-type: none"> • The VRF must be defined before it can be added to a list.
Step 5	Repeat Step 4 to add other VRFs to the list.	(Optional) If you want a trunk interface with one VRF, your list only needs one VRF.

	Command or Action	Purpose
Step 6	exit-vrf-list Example: <pre>Router(config-vrf-list)# exit-vrf-list</pre>	Exits VRF list configuration mode.
Step 7	interface <i>type number</i> Example: <pre>Router(config)# interface gigabitethernet 1/1/1</pre>	Configures an interface and enters interface configuration mode.
Step 8	vnet trunk list <i>vrf-list-name</i> Example: <pre>Router(config-if)# vnet trunk list mylist</pre>	Defines a trunk interface and enables the VRFs that are in the VRF list. <ul style="list-style-type: none"> • Use the <i>vrf-list-name</i> defined in Step 3.
Step 9	ip address <i>ip-address mask</i> Example: <pre>Router(config-if)# ip address 10.1.3.1 255.255.255.0</pre>	Sets a primary IP address for the interface.
Step 10	end Example: <pre>Router(config-if) end</pre>	Ends the configuration session and returns to privileged EXEC mode.
Step 11	show vrf list [<i>vrf-list-name</i>] Example: <pre>Router# show vrf list mylist</pre>	Displays information about a VRF list.

Configuring an EVN Edge Interface

Perform this task to configure an edge interface, which connects a user device to a virtual network. Traffic carried over an edge interface is untagged. The edge interface determines which virtual network the received traffic belongs to. Each edge interface is mapped to only one virtual network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface gigabitethernet 1/0/0	Configures an interface type and enters interface configuration mode.
Step 4	vrf forwarding <i>vrf-name</i> Example: Router(config-if)# vrf forwarding red	Defines an edge interface and determines the VRF that the incoming traffic belongs to. <ul style="list-style-type: none"> • The <i>vrf-name</i> must already be defined by a vrf definition command. • In this example, incoming traffic belongs to VRF red. <p>Note Make sure you are not on the trunk interface when you are trying to configure an edge interface.</p>
Step 5	ip address <i>ip-address mask</i> Example: Router(config-if)# ip address 10.1.1.1 255.255.255.0	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 6	end Example: Router(config-if) end	Ends the configuration session and returns to privileged EXEC mode.

What to Do Next

After you have configured an edge interface and a trunk interface, refer to your network diagram and log on to a different router. If it has an edge interface, configure that interface. If it has a trunk interface, configure that interface with the appropriate VRFs. Continue configuring each of the routers and interfaces that belong to each VRF.

Configure other protocol features you want running in your VRFs. See the appropriate IP Routing configuration guide.

Verifying EVN Configurations

Perform any of the following steps in this task to verify your configuration. Because a virtual network is a VRF, all the existing VRF **show** commands are supported for virtual networks. If a router has a mix of VRFs and virtual networks, the various **show vrf** commands will include both VRFs and virtual networks in the output.

SUMMARY STEPS

1. **enable**
2. **show vnet tag**
3. **show running-config [vrf | vnet] [vrf-name]**
4. **show vrf list [vrf-list-name]**
5. **show {vrf | vnet} [ipv4 | ipv6] [interface | brief | detail | lock] [vrf-name]**
6. **show {vrf | vnet} counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	show vnet tag Example: Router# show vnet tag	(Optional) Displays where each tag has been configured or used.
Step 3	show running-config [vrf vnet] [vrf-name] Example: Router# show running-config vrf green	(Optional) Displays the VRFs in the running configuration, displays the interfaces in the VRFs, and displays the protocol configurations for Multi-VRF.
Step 4	show vrf list [vrf-list-name] Example: Router# show vrf list	(Optional) Displays information about VRF lists, such as the VRFs in each list.
Step 5	show {vrf vnet} [ipv4 ipv6] [interface brief detail lock] [vrf-name] Example: Router# show vnet detail	(Optional) Displays information about the VRFs.
Step 6	show {vrf vnet} counters Example: Router# show vnet counters	(Optional) Displays information about the number of VRFs or virtual networks supported and configured.

Configuration Examples for Configuring EVN

Example: Virtual Networks Using OSPF with network Commands

In this example, **network** commands associate a shared VRF interface with a base VRF and two named VRFs, red and blue. There are three OSPF instances because each VRF needs its own OSPF instance. OSPF 1 has no VRF, so it is **vnet global**.

```
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

```

Example: Virtual Networks Using OSPF with ip ospf vnet area Command

This example differs from the prior example regarding the association between OSPF instances and a particular interface. In this example, OSPF is running on all of the virtual networks of a trunk interface. The **ip ospf vnet area** command associates the GigabitEthernet 0/0/0 interface with the three OSPF instances.

```

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

```

Example: Command Inheritance and Virtual Network Interface Mode Override in an EIGRP Environment

This example shows a GigabitEthernet interface configured with various EIGRP commands:

```

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

```

Because a trunk is configured, a VRF subinterface is automatically created and the commands on the main interface are inherited by the VRF subinterface (g0/0/0.3, where the number 3 is the tag number from vnet tag 3.)

```

R1# show running-config vrf red
Building configuration...
Current configuration : 1072 bytes
vrf definition red
 vnet tag 3
 !
 address-family ipv4
 exit-address-family
 !

```

If you display that hidden subinterface with the **show derived-config** command, you'll see that all of the commands entered on GigabitEthernet 0/0/0 have been inherited by GigabitEthernet 0/0/0.3:

```

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

```

You can override those commands by using virtual network interface mode (under the **vnet name** command). For example:

```

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
 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

```

Notice that g0/0.3 is now using the override settings:

```
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

```

Commands entered in **vnet name** submode are sticky. That is, when you enter a command in **vnet name** submode, it will nvgen, regardless of whether it is set to the same value as the default value. For example, the default hello value is 5. When the **ip hello eigrp** command is entered in **vnet name** submode, it will nvgen; it does not do that in any other mode.

```

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
!
```

Because of this sticky factor, to remove a configuration entry in **vnet name** submode, you typically must use the default form of that command. Some commands can also be removed using the **no** form; it depends on the command. Some commands use the **no** form to disable the command instead, such as the **authentication** and **summary-address** commands.

```

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
```

Example: Command Inheritance and Virtual Network Interface Mode Override in a Multicast Environment

The following example illustrates command inheritance and virtual network interface mode override in a multicast network. A trunk interface leverages the fact that configuration requirements from different VRFs will be similar over the same trunk interface. Eligible commands configured on the trunk interface are inherited by all VRFs running over the same interface.

In this example, IP multicast (PIM sparse mode) is configured on the trunk interface, which has several VRFs:

```

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
```

The user decides that he does not want IP multicast configured for VRF red on GigabitEthernet 0/1/0, so he uses the virtual network interface mode override. IP Multicast is disabled for VRF red only. The **no ip pim** command disables all modes of Protocol Independent Multicast (PIM), including sparse mode, dense mode, and sparse-dense mode, for VRF red.

```

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
```

Example: EVN Using IP Multicast

The following example configures PIM sparse mode and leverages Anycast RP for RP redundancy. In this example, only one VRF is configured.

The example shows how to enable multicast routing globally and on each L3 interface. The black text indicates the group of commands configuring the global table; the red text indicates the group of commands configuring VRF red.

```

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

```

Configure the RP in the VRF using Anycast RP.

```

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

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Information about Easy Virtual Network configuration tasks	“Overview of Easy Virtual Networks” module in the <i>Easy Virtual Network Configuration Guide</i>
Easy Virtual Network shared services and route replication configuration tasks	“Configuring Easy Virtual Network Shared Services” module in the <i>Easy Virtual Network Configuration Guide</i>
Easy Virtual Network management and troubleshooting	“Easy Virtual Network Management and Troubleshooting” module in the <i>Easy Virtual Network Configuration Guide</i>

MIBs

MIB	MIBs Link
<p>Any MIB that gives VRF information will continue to work with EVN. VRF-independent MIBs report information on every VRF in a system.</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

Feature Information for Configuring Easy Virtual Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Configuring Easy Virtual Network

Feature Name	Releases	Feature Information
EVN VNET Trunk	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>This module describes how to configure virtual IP networks. An EVN is an IP-based virtualization technology that provides end-to-end virtualization of the network. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other.</p> <p>The following commands were modified: vrf definition , vrf forwarding .</p> <p>The following commands were introduced: description (vrf definition submode), exit-if-vnet , exit-vrf-list , member (vrf list), routing-context , show running-config vnet , show vnet , show vnet counters , show vnet tag , show vrf counters , show vrf list , vnet , vnet tag , vnet trunk , vrf list .</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 provides Easy Virtual Network support for OSPF.</p> <p>The following commands were modified: ip ospf database-filter all out , ip ospf demand-circuit , ip ospf flood-reduction , ip ospf mtu-ignore , ip ospf shutdown .</p> <p>The following command was introduced: ip ospf vnet area .</p>

Feature Name	Releases	Feature Information
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 provides Easy Virtual Network support for EIGRP. The following commands were modified: ip summary-address eigrp , summary-metric .
EVN Multicast	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 Multicast provides Easy Virtual Network support for IP Multicast.



Easy Virtual Network Management and Troubleshooting

This module describes how to manage and troubleshoot Easy Virtual Network (EVN).

- [Finding Feature Information](#), page 43
- [Prerequisites for EVN Management and Troubleshooting](#), page 43
- [Information About EVN Management and Troubleshooting](#), page 44
- [How to Manage and Troubleshoot EVN](#), page 45
- [Additional References](#), page 50
- [Feature Information for EVN Management and Troubleshooting](#), page 51

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EVN Management and Troubleshooting

- Read the "Overview of Easy Virtual Network" section and the "Configuring Easy Virtual Network" section, and implement EVN.

Information About EVN Management and Troubleshooting

Routing Context for EXEC Mode Reduces Repetitive VRF Specification

There may be occasions when you want to issue several EXEC commands to apply to a single virtual network. In order to reduce the repetitive entering of virtual routing and forwarding (VRF) names for multiple EXEC commands, the **routing-context vrf** command allows you to set the VRF context of such EXEC commands once, and then proceed using EXEC commands.

The table below shows four EXEC commands in Cisco IOS XE software without routing context and in routing context. Note that in the left column, each EXEC command must specify the VRF. In the right column, the VRF context is specified once and the prompt changes to reflect that VRF; there is no need to specify the VRF in each command.

Table 5: EXEC Commands Routing Context

EXEC Commands CLI without Routing Context	EXEC Routing Context
—	Router# routing-context vrf red Router%red#
Router# show ip route vrf red [Routing table output for VRF red]	Router%red# show ip route [Routing table output for VRF red]
Router# ping vrf red 10.1.1.1 [Ping result using VRF red]	Router%red# ping 10.1.1.1 [Ping result using VRF red]
Router# telnet 10.1.1.1 /vrf red [Telnet to 10.1.1.1 in VRF red]	Router%red# telnet 10.1.1.1 [Telnet to 10.1.1.1 in VRF red]
Router# traceroute vrf red 10.1.1.1 [Traceroute output in VRF red]	Router%red# traceroute 10.1.1.1 [Traceroute output in VRF red]

Output of traceroute Command Indicates VRF Name and VRF Tag

Output of the **traceroute** command is enhanced to make troubleshooting easier by displaying the incoming VRF name/tag and the outgoing VRF name/tag, as shown in the following example:

```
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 * * *

```

Debug Output Filtering Per VRF

Using EVN, you can filter debug output per VRF by using the **debug condition vrf** command. The following is sample output from the **debug condition vrf** command:

```

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#

```



Note

The **debug condition vrf** command does not work in an EIGRP environment.

CISCO-VRF-MIB

EVN provides a CISCO-VRF-MIB for VRF discovery and management.

How to Manage and Troubleshoot EVN

Setting the Routing Context for EXEC Mode to a Specific VRF

To reduce the repeated entering of virtual routing and forwarding (VRF) names when you are issuing EXEC commands on a router, set the routing context of the EXEC commands once, and then proceed with entering them in any order. Perform this task to set the routing context for EXEC mode to a specific VRF, issue EXEC commands, and then restore the system to the global EXEC context.

SUMMARY STEPS

1. **enable**
2. **routing-context vrf** *vrf-name*
3. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **static download**]
4. **ping** [*protocol* [**tag**] {*host-name* | *system-address*}]
5. **telnet** *host* [*port*]
6. **traceroute** [**vrf** *vrf-name* | **topology** *topology-name*] [*protocol*] *destination*
7. **routing-context vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	routing-context vrf <i>vrf-name</i> Example: <pre>Router# routing-context vrf red</pre>	Enters the routing context for EXEC mode to a specified VRF.
Step 3	show ip route [<i>ip-address</i> [<i>mask</i>] [longer-prefixes] <i>protocol</i> [<i>process-id</i>] static download] Example: <pre>Router%red# show ip route</pre>	(Optional) Displays the current state of the routing table. <ul style="list-style-type: none"> • The system prompt changes to reflect the target VRF. • This example shows the show ip route command issued within the context of vNET red. The routing table for vNET red would be displayed.
Step 4	ping [<i>protocol</i> [tag] { <i>host-name</i> <i>system-address</i> }] Example: <pre>Router%red# ping 10.1.1.1</pre>	(Optional) Sends an echo request packet to an address. <ul style="list-style-type: none"> • This example shows the ping command issued within the context of vNET red. Ping results using vNET red would be displayed.
Step 5	telnet <i>host</i> [<i>port</i>] Example: <pre>Router%red# telnet 10.1.1.1</pre>	(Optional) Logs in to a host that supports Telnet.
Step 6	traceroute [vrf <i>vrf-name</i> topology <i>topology-name</i>] [<i>protocol</i>] <i>destination</i> Example: <pre>Router%red# traceroute 10.1.1.1</pre>	(Optional) Displays the route that packets will take to the destination.
Step 7	routing-context vrf global Example: <pre>Router%red# routing-context vrf global</pre> Example: <pre>Router></pre>	(Optional) Restores the system to the global EXEC context. <ul style="list-style-type: none"> • The prompt returns to the user EXEC prompt.

Enabling Debug Output for VRFs

SUMMARY STEPS

1. `enable`
2. `debug vrf {create | delete | error | ha | initialization | interface | ipv4 | ipv6 | issu | lock | lookup | mpls | selection}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>debug vrf {create delete error ha initialization interface ipv4 ipv6 issu lock lookup mpls selection}</code></p> <p>Example:</p> <pre>Router# debug vrf ipv4</pre>	<p>Displays VRF debugging information.</p>

Setting SNMP v2c Context for Virtual Networks

Perform this task to map an SNMP v2c context to a VRF. The following SNMP v2c configurations will then be done by the system automatically:

- Context creation (instead of the `snmp-server context` command), using the same name as the *context-name* entered in the `snmp context` command.
- Group creation (instead of the `snmp-server group` command), using the same name as the *community-name* entered in the `snmp context` command.
- Community creation (instead of the `snmp-server community` command), using the same name as the *community-name* entered in the `snmp context` command. The default permission is `ro` (read-only).
- Community context mapping (instead of the `snmp mib community-map` command).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv4**
5. **snmp context** *context-name* [**community** *community-name* [**rw** | **ro**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition vrf1	Defines a virtual routing and forwarding instance (VRF) and enters VRF configuration mode.
Step 4	address-family ipv4 Example: Device(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 5	snmp context <i>context-name</i> [community <i>community-name</i> [rw ro]] Example: Router(config-vrf)# snmp context xxx community yyy	Sets the SNMP v2c context for the VRF. <ul style="list-style-type: none"> • The default is read-only (ro).

Setting SNMP v3 Context for Virtual Networks

Perform this task to map an SNMP v3 context to a virtual routing and forwarding (VRF). The following SNMP v3 configurations will then be done by the system automatically:

- Context creation (instead of the **snmp-server context** command), using the same name as the *context-name* entered in the **snmp context** command.
- Group creation (instead of the **snmp-server group** command). The group name will be generated by appending “_acnf” to the *context-name* entered in the **snmp context** command.
- User creation (instead of the **snmp-server user** command). The user will be created using the details configured in the **snmp context** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv4**
5. **snmp context** *context-name* [**user** *username* [**credential** | [**encrypted**] [**auth** {**md5** *password* | **sha** *password*}] [**access** {*access-list-number* | *access-list-name* | **ipv6** *access-list-name*}]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition vrf1	Defines a VRF and enters VRF configuration mode.
Step 4	address-family ipv4 Example: Device(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 5	snmp context <i>context-name</i> [user <i>username</i> [credential [encrypted] [auth { md5 <i>password</i> sha <i>password</i> }] [access { <i>access-list-number</i> <i>access-list-name</i> ipv6 <i>access-list-name</i> }]]]	Sets the SNMP v3 context for the VRF.

	Command or Action	Purpose
	<p>Example:</p> <pre>Router(config-vrf)# snmp context green_ctx user green_comm encrypted</pre>	

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Overview of Easy Virtual Network	“Overview of Easy Virtual Network” module in the <i>Easy Virtual Network Configuration Guide</i>
Configuring Easy Virtual Network	“Configuring Easy Virtual Network” module in the <i>Easy Virtual Network Configuration Guide</i>
Easy Virtual Network shared services and route replication	“Easy Virtual Network Shared Services” module in the <i>Easy Virtual Network Configuration Guide</i>

MIBs

MIB	MIBs Link
<p>Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system:</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for EVN Management and Troubleshooting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for EVN Management and Troubleshooting

Feature Name	Releases	Feature Information
EVN Cisco EVN MIB	15.0(1)SY 15.1(1)SG 15.3(2)T	EVN Cisco EVN MIB simplifies SNMP configuration. The following command was modified: snmp context . .
EVN Traceroute	15.0(1)SY 15.1(1)SG 15.3(2)T	EVN Traceroute enhances output of the traceroute command to display the VRF name and tag. The following command was modified: traceroute . .
EVN VNET Trunk	15.0(1)SY 15.2(1)SY 15.1(1)SG 15.3(2)T 15.4(1)T	Users can filter debug output per VRF by using the debug condition vrf command. The following commands were introduced: debug condition vrf , debug vrf .



Configuring Easy Virtual Network Shared Services

This chapter describes how to use route replication and redistribution to share services in an Easy Virtual Network (EVN).

- [Finding Feature Information, page 53](#)
- [Prerequisites for Virtual IP Network Shared Services, page 53](#)
- [Restrictions for Virtual IP Network Shared Services, page 54](#)
- [Information About Easy Virtual Network Shared Services, page 54](#)
- [How to Share Services Using Easy Virtual Network , page 57](#)
- [Configuration Example for Easy Virtual Network Shared Services, page 67](#)
- [Additional References, page 72](#)
- [Feature Information for Easy Virtual Network Shared Services, page 73](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Virtual IP Network Shared Services

- Read the “Overview of Easy Virtual Networks” module.
- Implement EVN based on the “Configuring Easy Virtual Networks” module.

Restrictions for Virtual IP Network Shared Services

Route replication is supported for Static, Enhanced Interior Gateway Routing Protocol (EIGRP), and Open Shortest Path First (OSPF) routes. It is not possible to replicate routes to and from Border Gateway Protocol (BGP), but that is not an issue because the BGP import and export method of copying routes between Virtual Routing and Forwarding (VRF) is available in a virtual network.

Information About Easy Virtual Network Shared Services

Shared Services in an Easy Virtual Network

There are some common services (such as database and application servers) that multiple virtual networks need to access. Sharing these services are beneficial because:

- They are usually not duplicated for each group.
- It is economical, efficient, and manageable.
- Policies can be centrally deployed.

To achieve route separation, you could replicate the service, either physically or virtually, one service for each virtual network. However, that solution might not be cost effective or feasible. For a router that supports EVN, the solution is to perform route replication and route redistribution.

Route replication allows shared services because routes are replicated between virtual networks and clients who reside in one virtual network can reach prefixes that exist in another virtual network.

A shared services approach works best for Dynamic Name Systems (DNS), Dynamic Host Configuration Protocol (DHCP), and corporate communications. It is not a solution for sharing access to an Internet gateway.

Easy Virtual Network Shared Services Easier than VRF-Lite

Sharing servers in VRF-Lite requires route distinguishers (RDs), route targets with importing and exporting, and configuring BGP.

In an EVN environment, shared services are achieved with route replication, which is a simple deployment. Route replication requires no BGP, no RD, no route targets, and no import or export.

In summary, the BGP import and export method of copying routes between VRFs works with both VRF-Lite and EVN. However, route replication is the simpler alternative to enable sharing of common services across multiple virtual networks.

Route Replication Process in Easy Virtual Network

With shared services, clients and servers are located in different virtual networks. To achieve connectivity between clients and servers, routes must be exchanged among virtual networks. Depending on whether VRF-Lite or EVN is implemented, route exchanges among VRFs are accomplished in one of the following ways:

- If VRF-Lite is implemented, route leaking is achieved via BGP by using the route import/export feature.
- If EVN is implemented, route replication is supported directly by the Routing Information Base (RIB); there is no dependency on BGP. After routes are replicated from a different virtual network, those routes are propagated across each virtual network through existing redistribution into the Interior Gateway Protocol (IGP).

In the following route replication scenario, a router has two VRFs named Services and User-A. OSPF is configured:

```
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
```

Furthermore, route replication is configured for VRF User-A:

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

In the scenario, the following RIB for the VRF Services contains four routes, three of which are replicated to the RIB for VRF User-A. Route replication creates a link to the source RIB, as shown in the figure below.

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



Configuring route replication allows mutual redistribution between virtual IP networks. In the case of shared services, you configure route replication within the VRF that needs access to shared services. Within each **route-replicate** command, you can optionally filter out routes with a route map to prevent a routing loop. That is, you do not want to redistribute routes back into the original routing protocol. You do not want a native route to show up as a replicated route.

Where to Implement Route Replication

We recommend implementing route replication on the router as close to the shared service as possible. Ideally, the router that is directly connected to the server subnet should be used, to eliminate the need to redistribute the host prefixes on the server VRF, and, thereby, avoid a potential routing loop.

Route Replication Behavior for Easy Virtual Network

This section describes the behavior of route replication for EVN, which differs from the behavior for Multi-Topology Routing. In an EVN environment:

- The **route-replicate** command is accepted only under the **address-family ipv4** command, which is configured under the **vrf definition** command.
- The **route-replicate** command replicates routes into the base topology within the specified address family.
- If **all** is specified as a source protocol, only one **route-replicate** command is allowed per VRF for a given destination topology.
- The **no route-replicate** command is allowed to exclude a source protocol.
- If **all** is specified as a source protocol, then connected routes are replicated (unlike in the Multi-Topology Routing version of the **route-replicate** command).
- A replicated route inherits the administrative distance and source protocol of the source route.

Route Preference Rules After Route Replication in Easy Virtual Network

If a route is replicated, the following rule determines route preference:

- If two routes are owned by the same protocol and have the same source VRF, and if one of the routes is NOT replicated, then the nonreplicated route is preferred.

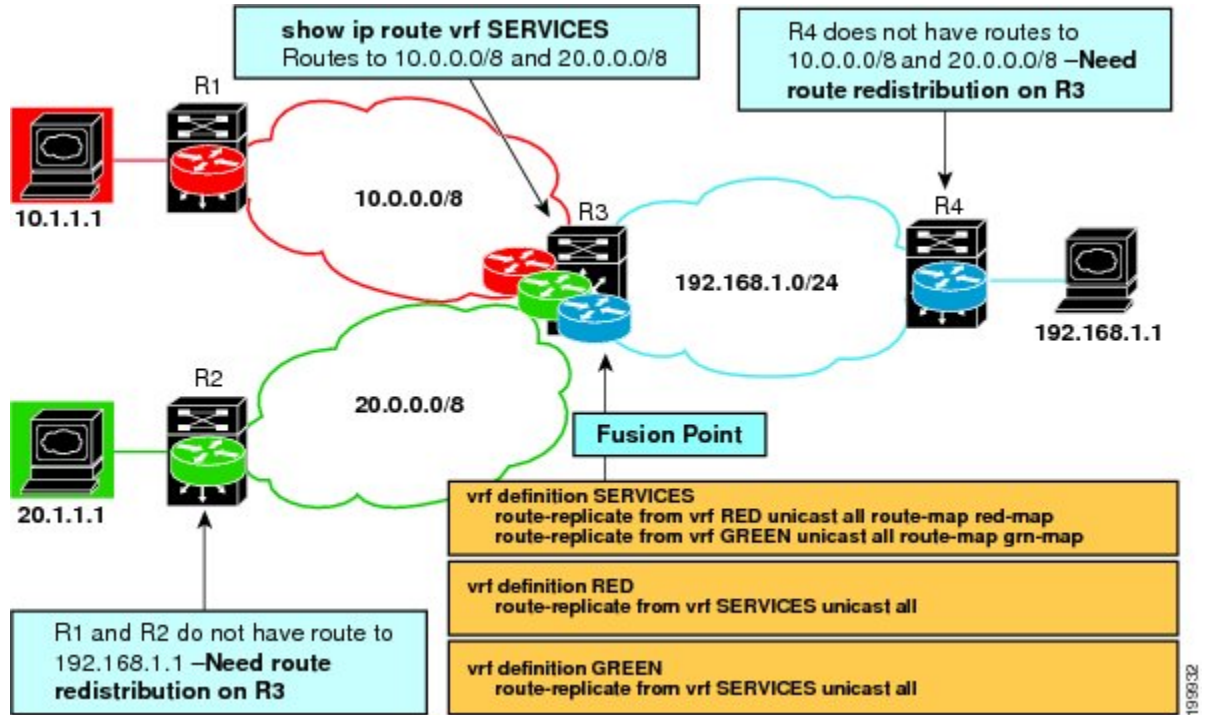
If the above rule does not apply, the following rules determine route preference, in this order:

- 1 Prefer the route with smaller administrative distance.
- 2 Prefer the route with smaller default administrative distance.
- 3 Prefer a non-replicated route over a replicated route.
- 4 Compare original vrf-names. Prefer the route with the lexicographically smaller vrf-name.
- 5 Compare original sub-address-families: Prefer unicast over multicast.
- 6 Prefer the oldest route.

How to Share Services Using Easy Virtual Network

Configuring Route Replication to Share Services in Easy Virtual Network

Perform this task to replicate routes from one VRF to another. The examples in the task table are based on the figure below.



In this particular task, routes from VRF SERVICES are replicated to both VRF RED and VRF GREEN, and VRF RED and VRF GREEN are not allowed to share routes between them. In order to allow bidirectional traffic, routes from VRF RED and VRF GREEN are also replicated to VRF SERVICES.



Note

In a real EVN environment, there would also be route replication between VRF SERVICES and a third VRF, and maybe more VRFs. Such replication is left out of the following configuration task for the sake of brevity.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *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. **no shutdown**
19. **exit**
20. **router ospf** *process-id vrf vrf-name*
21. **network** *ip-address wildcard-mask area area-id*
22. **exit**
23. **router ospf** *process-id [vrf vrf-name]*
24. **network** *ip-address wildcard-mask area area-id*
25. **exit**
26. **vrf definition** *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. **vrf definition** *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 vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition SERVICES	Defines a VRF and enters VRF configuration mode.
Step 4	vnet tag <i>number</i> Example: Router(config-vrf)# vnet tag 100	Specifies the global, numeric tag for the VRF. <ul style="list-style-type: none"> • The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 5	description <i>string</i> Example: Router(config-vrf)# description shared services	(Optional) Describes a VRF to help the network administrator looking at the configuration file.
Step 6	address-family ipv4 Example: Router(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	exit Example: Router(config-vrf-af)# exit	Exits to VRF configuration mode.
Step 8	exit Example: Router(config-vrf)# exit	Exits to global configuration mode.

	Command or Action	Purpose
Step 9	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition RED	Defines a VRF and enters VRF configuration mode.
Step 10	vnet tag <i>number</i> Example: Router(config-vrf)# vnet tag 200	Specifies the global, numeric tag for the VRF. <ul style="list-style-type: none"> The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 11	description <i>string</i> Example: Router(config-vrf)# description user of services	(Optional) Describes a VRF to help the network administrator looking at the configuration file.
Step 12	address-family ipv4 Example: Router(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 13	exit Example: Router(config-vrf-af)# exit	Exits to VRF configuration mode.
Step 14	exit Example: Router(config-vrf)# exit	Exits to global configuration mode.
Step 15	interface <i>type number</i> Example: Router(config)# interface gigabitethernet 0/0/0	Configures an interface type and number and enters interface configuration mode.
Step 16	vrf forwarding <i>vrf-name</i> Example: Router(config-if)# vrf forwarding SERVICES	Associates a VRF instance with an interface.

	Command or Action	Purpose
Step 17	<p>ip address <i>ip-address mask</i></p> <p>Example:</p> <pre>Router(config-if)# ip address 192.168.1.3 255.255.255.0</pre>	Sets a primary IP address for an interface.
Step 18	<p>no shutdown</p> <p>Example:</p> <pre>Router(config-if)# no shutdown</pre>	Restarts an interface.
Step 19	<p>exit</p> <p>Example:</p> <pre>Router(config-if)# exit</pre>	Exits to global configuration mode.
Step 20	<p>router ospf <i>process-id vrf vrf-name</i></p> <p>Example:</p> <pre>Router(config)# router ospf 99 vrf SERVICES</pre>	<p>Configures an OSPF routing process and enters router configuration mode.</p> <ul style="list-style-type: none"> • This example uses OSPF; EIGRP is also available.
Step 21	<p>network <i>ip-address wildcard-mask area area-id</i></p> <p>Example:</p> <pre>Router(config-router)# network 192.168.1.0 0.0.0.255 area 0</pre>	Defines the interfaces on which OSPF runs and the area ID for those interfaces.
Step 22	<p>exit</p> <p>Example:</p> <pre>Router(config-router)# exit</pre>	Exits to global configuration mode.
Step 23	<p>router ospf <i>process-id [vrf vrf-name]</i></p> <p>Example:</p> <pre>Router(config)# router ospf 98 vrf RED</pre>	Configures an OSPF routing process and enters router configuration mode.
Step 24	<p>network <i>ip-address wildcard-mask area area-id</i></p> <p>Example:</p> <pre>Router(config-router)# network 192.168.1.0 0.0.0.255 area 0</pre>	Defines the interfaces on which OSPF runs and the area ID for those interfaces.

	Command or Action	Purpose
Step 25	exit Example: <pre>Router(config-router)# exit</pre>	Exits to the global configuration mode.
Step 26	vrf definition <i>vrf-name</i> Example: <pre>Router(config)# vrf definition RED</pre>	Defines a VRF and enters VRF configuration mode.
Step 27	address-family ipv4 Example: <pre>Router(config-vrf)# address-family ipv4</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 28	route-replicate from [vrf <i>vrf-name</i>] { multicast unicast } { all <i>protocol-name</i> } [route-map <i>map-tag</i>] Example: <pre>Router(config-vrf-af)# route replicate from vrf SERVICES unicast all</pre>	Replicates routes into the base topology within the specified address family. <ul style="list-style-type: none"> • If the all keyword is specified as a source protocol, only one route-replicate command is allowed per VRF for a given destination topology. • Use the connected keyword as a source <i>protocol-name</i> in order to replicate only connected routes.
Step 29	exit Example: <pre>Router(config-vrf-af)# exit</pre>	Exits to VRF configuration mode.
Step 30	exit Example: <pre>Router(config-vrf)# exit</pre>	Exits to global configuration mode.
Step 31	vrf definition <i>vrf-name</i> Example: <pre>Router(config)# vrf definition SERVICES</pre>	Defines a VRF and enters VRF configuration mode.
Step 32	address-family ipv4 Example: <pre>Router(config-vrf)# address-family ipv4</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
Step 33	route-replicate from [<i>vrf vrf-name</i>] { multicast unicast } { all <i>protocol-name</i> } [route-map <i>map-tag</i>] Example: Router(config-vrf-af)# route replicate from vrf RED unicast all	Replicates routes into the base topology within the specified address family. <ul style="list-style-type: none"> This is the reciprocal replication to Step 28 to allow bidirectional traffic.
Step 34	end Example: Router(config-vrf-af)# end	Exits configuration mode.
Step 35	show ip route vrf <i>vrf-name</i> Example: Router# show ip route vrf RED	(Optional) Displays routes, including those replicated, which are indicated by a plus sign (+).

Example

The following is sample output from the **show ip route vrf** command based on the task in the preceding task table:

```
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#
```

What to Do Next

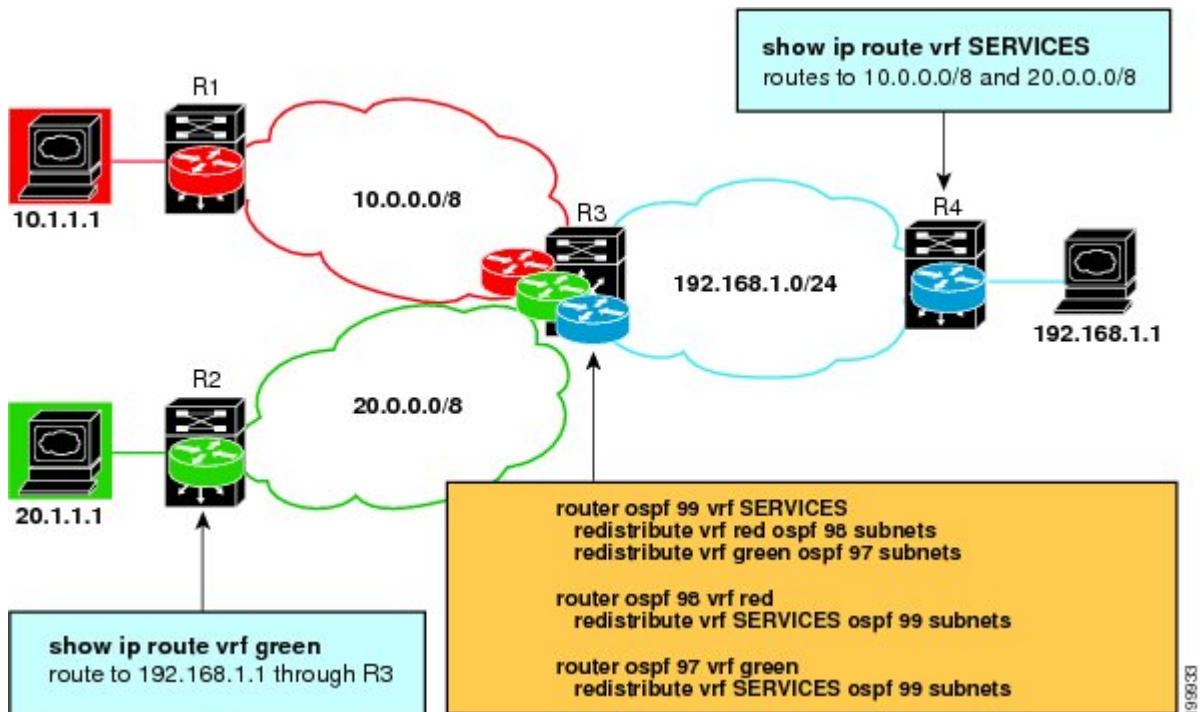
After you perform the “Configuring Route Replication to Share Services in Easy Virtual Network” task, you must configure VRF GREEN as per the figure above, noting that Router 3 has routes to 10.0.0.0/8 and 20.0.0.0/8 and Router 1 and Router 2 have a route to 192.168.1.0/24.

After the configuration is complete, Router 1 and Router 2 still do not have a route to the shared service residing on 192.168.1.1 and Router 4 does not have routes to 10.0.0.0/8 and 20.0.0.0/8. Such access requires the route redistribution performed in the next task, "Configuring Redistribution to Share Services in EVN".

Configuring Redistribution to Share Services in Easy Virtual Network

This task is based on the assumption that you also performed the task, [Configuring Route Replication to Share Services in EVN](#).

The figure below shows the same networks we used in the figure above. In this task, we perform redistribution on Router 3 so that Router 1 and Router 2 have a route to the shared service residing on 192.168.1.1.



SUMMARY STEPS

1. enable
2. configure terminal
3. **router ospf** *process-id* **vrf** *vrf-name*
4. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
5. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
6. exit
7. **router ospf** *process-id* **vrf** *vrf-name*
8. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
9. exit
10. **router ospf** *process-id* **vrf** *vrf-name*
11. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
12. end
13. **show ip route** **vrf** *vrf-name*

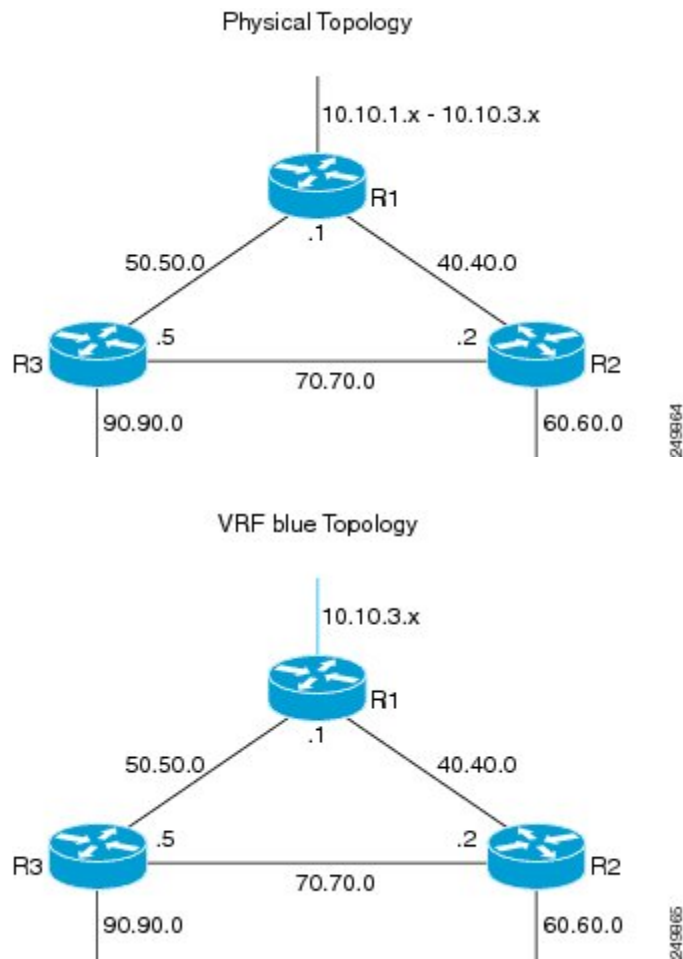
DETAILED STEPS

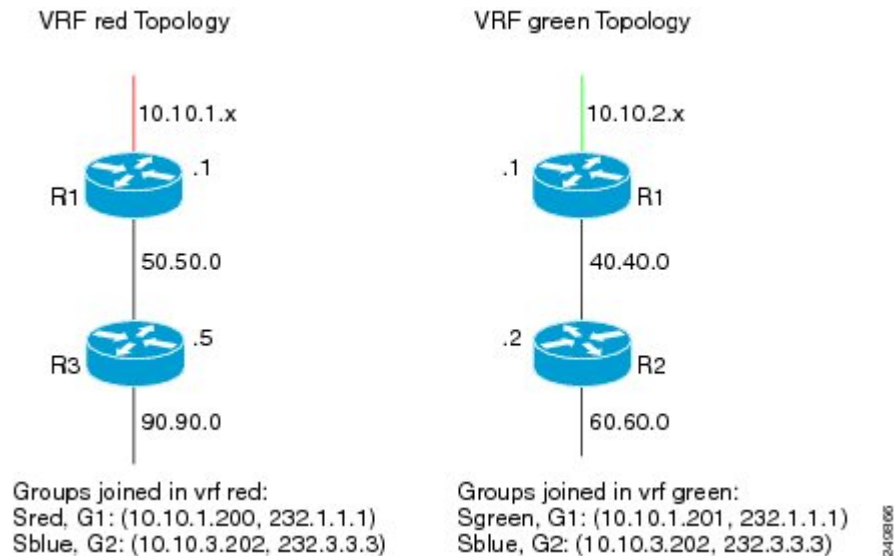
	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router ospf <i>process-id</i> vrf <i>vrf-name</i> Example: Router(config)# router ospf 99 vrf SERVICES	Configures an OSPF routing process and enters router configuration mode.
Step 4	redistribute vrf <i>vrf-name</i> ospf <i>process-id</i> subnets Example: Router(config-router)# redistribute vrf RED ospf 98 subnets	Redistributes routes from one routing domain into another routing domain.
Step 5	redistribute vrf <i>vrf-name</i> ospf <i>process-id</i> subnets Example: Router(config-router)# redistribute vrf GREEN ospf 97 subnets	Redistributes routes from one routing domain into another routing domain.
Step 6	exit Example: Router(config-router)# exit	Exits to global configuration mode.
Step 7	router ospf <i>process-id</i> vrf <i>vrf-name</i> Example: Router(config)# router ospf 98 vrf RED	Configures an OSPF routing process and enters router configuration mode.
Step 8	redistribute vrf <i>vrf-name</i> ospf <i>process-id</i> subnets Example: Router(config-router)# redistribute vrf SERVICES ospf 99 subnets	Redistributes routes from one routing domain into another routing domain.

	Command or Action	Purpose
Step 9	exit Example: Router(config-router)# exit	Exits to global configuration mode.
Step 10	router ospf process-id vrf vrf-name Example: Router(config)# router ospf 97 vrf GREEN	Configures an OSPF routing process and enters router configuration mode.
Step 11	redistribute vrf vrf-name ospf process-id subnets Example: Router(config-router)# redistribute vrf SERVICES ospf 99 subnets	Redistributes routes from one routing domain into another routing domain.
Step 12	end Example: Router(config-router)# end	Exits configuration mode.
Step 13	show ip route vrf vrf-name Example: Router# show ip route vrf RED	(Optional) Displays routes, including those replicated, which are indicated by a plus sign (+).

Configuration Example for Easy Virtual Network Shared Services

Example: Easy Virtual Network Route Replication and Route Redistribution in a Multicast Environment





In the figures above there are three multicast streams:

- Sred, G1: (10.10.1.200, 232.1.1.1)--Source and receivers in VRF red
- Sgreen, G1: (10.10.2.201, 232.1.1.1)--Source and receivers in VRF green
- Sblue, G2: (10.10.3.202, 232.3.3.3)--Source in blue and receivers in VRFs red and green.

The server-prefix in VRF blue (10.10.3.0/24) is replicated and distributed into VRFs red and green on R3 and R2.

Multicast group 232.3.3.3 with its source in VRF blue has receivers in both VRF red and VRF green. The stream is transmitted over the shared VRF (blue), and then replicated into VRF red on R3 and into VRF green on R2.

R1 Configuration

```
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 Configuration

```

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 Configuration

```

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
!

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Overview of Easy Virtual Network	“Overview of Easy Virtual Network” module in the <i>Easy Virtual Network Configuration Guide</i>
Configuring Easy Virtual Network	“Configuring Easy Virtual Network” module in the <i>Easy Virtual Network Configuration Guide</i>
Easy Virtual Network management and troubleshooting	“Easy Virtual Network Management and Troubleshooting” module in the <i>Easy Virtual Network Configuration Guide</i>

MIBs

MIB	MIBs Link
<p>Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system:</p> <ul style="list-style-type: none"> • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

Feature Information for Easy Virtual Network Shared Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for Easy Virtual Network Shared Services

Feature Name	Releases	Feature Information
EVN Route Replication	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	This module describes how to use route replication and redistribution to share services in an EVN environment. This feature modifies the following command: redistribute (IP) This feature introduces the following command: route-replicate (VRF address family)