

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Configure](#)

[Network Diagram](#)

[Configurations](#)

[Verify](#)

[Inheritance with EIGRP Named mode](#)

[Route Replication with EIGRP name mode](#)

[Routing Context](#)

[Enhanced traceroute](#)

[Conclusion](#)

[References](#)

Introduction

The purpose of this document is to demonstrate the configuration of EVN (Easy Virtual Network) using EIGRP (Enhanced Interior Gateway Routing Protocol) Named mode. It is a supplement to the [Easy Virtual Network Configuration](#) document, which demonstrates the use of OSPF (Open Shortest Path First), as well as other advanced topics like VNET trunk lists and route replication. The EVN VNET was intended for operators to have an easier option than MPLS (Multi Protocol Label Switching) VPN (Virtual Private Network) or VRF-lite (Virtual Routing and Forwarding) for deploying multiple VRF's. EVN VNET uses a concept of cloned configuration for the routing protocols and VNET trunk interface to remove the burden from the operator and save some of the repetitive tasks. Troubleshooting EIGRP, routing or CEF (Cisco Express Forwarding) is outside of the scope of this document and unless noted you can follow normal troubleshooting procedures.

Prerequisites

Requirements

Cisco recommends that you have basic knowledge of EIGRP.

This feature is available in few releases after IOS version 15.2. To verify if EIGRP named mode with EVN VNET's is supported, check output of **show ip eigrp plugins**. If Easy Virtual Network version 1.00.00 or later is present, then your version supports this feature.

```
R1#show eigrp plugins
EIGRP feature plugins:::
eigrp-release : 21.00.00 : Portable EIGRP Release
: 1.00.10 : Source Component Release(rel21)
parser : 2.02.00 : EIGRP Parser Support
igrp2 : 2.00.00 : Reliable Transport/Dual Database
bfd : 2.00.00 : BFD Platform Support
mtr : 1.00.01 : Multi-Topology Routing (MTR)
```

eigrp-pfr : 1.00.01 : Performance Routing Support
EVN/vNets : 1.00.00 : Easy Virtual Network (EVN/vNets)
ipv4-af : 2.01.01 : Routing Protocol Support
ipv4-sf : 1.02.00 : Service Distribution Support
vNets-parse : 1.00.00 : EIGRP vNets Parse Support
ipv6-af : 2.01.01 : Routing Protocol Support
ipv6-sf : 2.01.00 : Service Distribution Support
snmp-agent : 2.00.00 : SNMP/SNMPv2 Agent Support

Note: EIGRP named mode with EVN VNETs is not supported in 15.1SY. In this version you must use classic mode EIGRP configuration which is already demonstrated in the available documentation.

BFD (

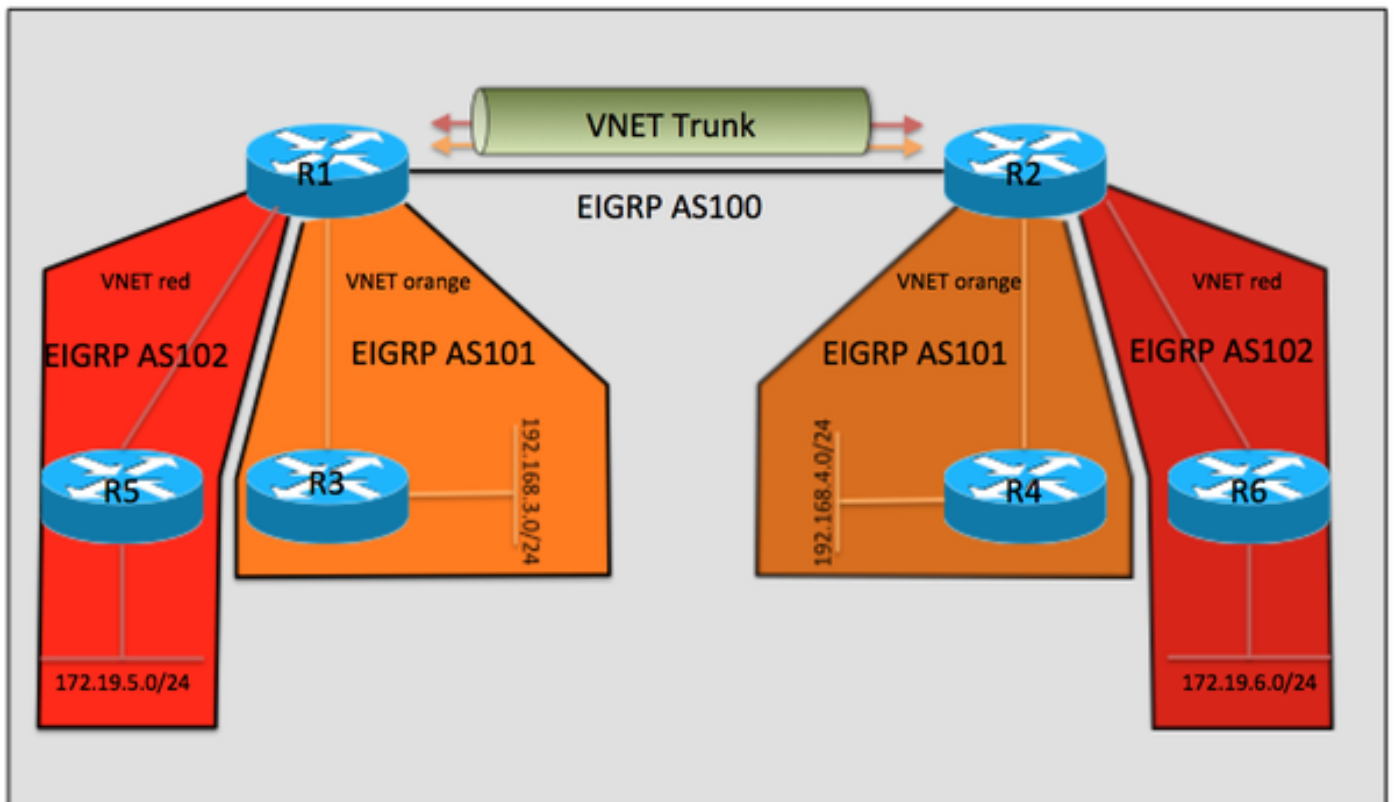
It is not advised to use af-interface default when using EIGRP named mode with EVN VNETs due to possible unpredictable inheritance.

Components Used

The information in this document was created from the devices in a specific lab environment running Cisco IOS version 15.6(1)S2. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Configure

Network Diagram



Configurations

The configurations of R3, R4, R5 and R6 are all similar, and therefore left out of the document. They are simply configured to form an EIGRP neighbor with R1 or R2, and they are not aware of the EVN VNET's used between R1 and R2.

Relevant configuration from R1

```
vrf definition orange
vnet tag 101
!
address-family ipv4
exit-address-family
!
vrf definition red
vnet tag 102
!
address-family ipv4
exit-address-family
!
interface Ethernet0/0
vnet trunk
ip address 10.12.12.1 255.255.255.0
!
interface Ethernet1/0
vrf forwarding orange
ip address 192.168.13.1 255.255.255.0
!
interface Ethernet2/0
vrf forwarding red
ip address 192.168.15.1 255.255.255.0
!
!
router eigrp named
!
address-family ipv4 unicast autonomous-system 100
!
af-interface Ethernet0/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
exit-address-family
!
address-family ipv4 unicast vrf orange autonomous-system 101
!
af-interface Ethernet1/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
network 192.168.13.0
exit-address-family
!
address-family ipv4 unicast vrf red autonomous-system 102
!
topology base
```

```
exit-af-topology
network 10.0.0.0
  network 192.168.15.0
exit-address-family
```

Relevant configuration from R2

```
vrf definition orange
  vnet tag 101
  !
  address-family ipv4
  exit-address-family
!
vrf definition red
  vnet tag 102
  !
  address-family ipv4
  exit-address-family
!
interface Ethernet0/0
  vnet trunk
ip address 10.12.12.2 255.255.255.0
!
interface Ethernet1/0
  vrf forwarding orange
  ip address 192.168.24.2 255.255.255.0
!
interface Ethernet2/0
  vrf forwarding red
  ip address 192.168.26.2 255.255.255.0
!
!
router eigrp named
  !
  address-family ipv4 unicast autonomous-system 100
!
af-interface Ethernet0/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
  network 10.0.0.0
exit-address-family
!
  address-family ipv4 unicast vrf orange autonomous-system 101
!
af-interface Ethernet1/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
  network 192.168.24.0
exit-address-family
!
  address-family ipv4 unicast vrf red autonomous-system 102
!
topology base
exit-af-topology
network 10.0.0.0
  network 192.168.26.0
exit-address-family
```

Verify

One of the benefits of Easy Virtual Network is the simplicity of the configuration. This is achieved by automatically configuring the VNET trunks for each VNET tag. Comparing EVN with VRF-lite, each sub-interface would need to be manually configured. Ethernet0/0 is the VNET trunk which connects R1 and R2, and a VNET sub-interface is automatically created for each VNET to meet the traffic separation requirements for EVN by appending frames with a dot1Q VNET tag. These sub-interfaces are not visible in the output of show running-configuration, however they can be seen with show derived-config.

```
R1#show derived-config | sec Ethernet0/0
interface Ethernet0/0
  vnet trunk
ip address 10.12.12.1 255.255.255.0
no ip redirects
no ip proxy-arp
interface Ethernet0/0.101
  description Subinterface for VNET orange
  encapsulation dot1Q 101
  vrf forwarding orange
  ip address 10.12.12.1 255.255.255.0
  no ip proxy-arp
interface Ethernet0/0.102
  description Subinterface for VNET red
  encapsulation dot1Q 102
  vrf forwarding red
  ip address 10.12.12.1 255.255.255.0
  no ip proxy-arp
```

Similarly, you can see that EIGRP configuration is also created automatically:

```
R1#show derived-config | sec router eigrp
router eigrp named
!
address-family ipv4 unicast autonomous-system 100
!
af-interface Ethernet0/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
exit-address-family
!
address-family ipv4 unicast vrf orange autonomous-system 101
!
af-interface Ethernet0/0.101
authentication mode hmac-sha-256 cisco
exit-af-interface
!
af-interface Ethernet1/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
network 192.168.13.0
exit-address-family
!
```

```

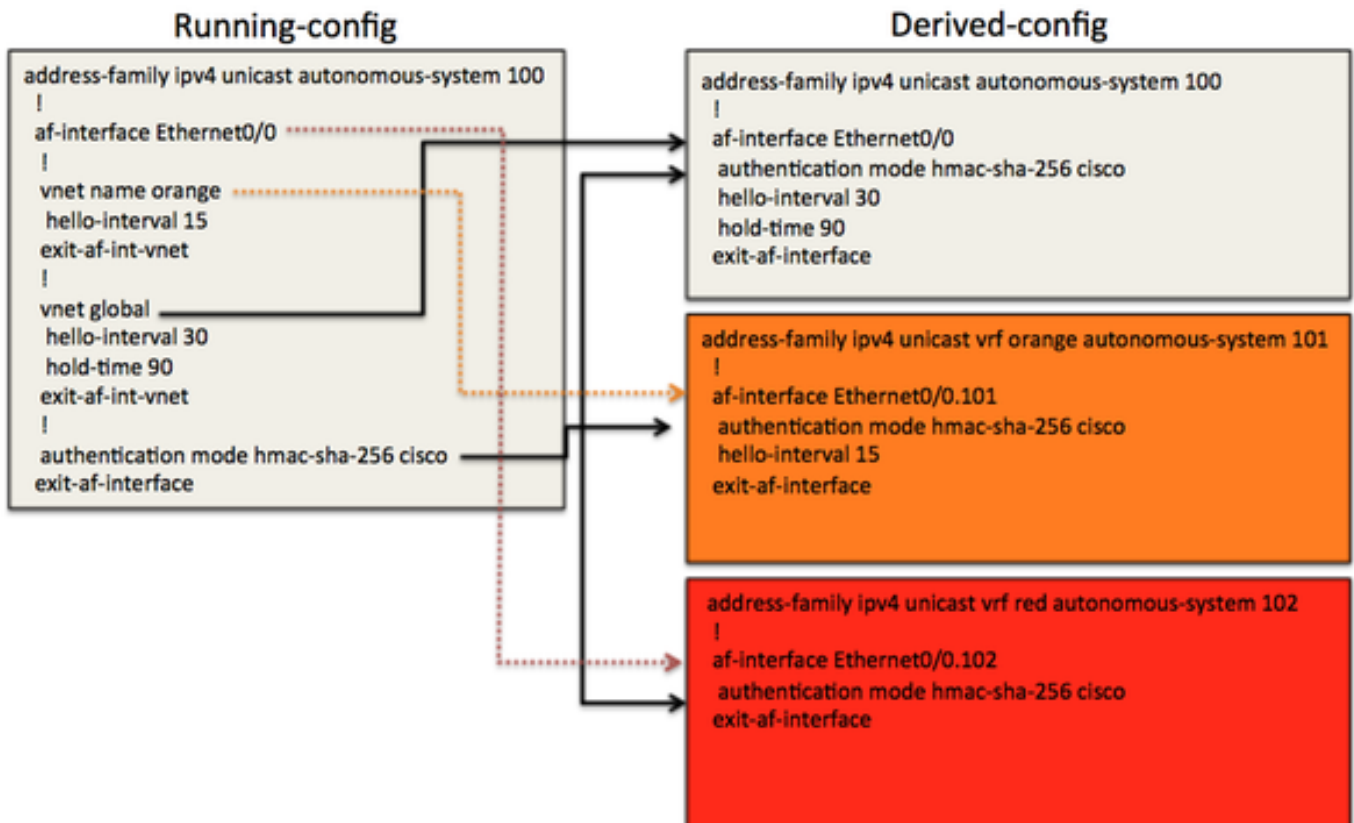
address-family ipv4 unicast vrf red autonomous-system 102
!
af-interface Ethernet0/0.102
 authentication mode hmac-sha-256 cisco
 exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
network 192.168.15.0
exit-address-family
R1#

```

An interesting observation in the output above is the af-interface inheritance for the VNET sub-interfaces from af-interface ethernet0/0 in the global vrf autonomous-system 100. Following section explains this in more details:

Inheritance with EIGRP Named mode

The figure below will be used to help visualize the inheritance rules when using EIGRP named mode with EVN VNETs.



In the example above there is a VNET trunk af-interface ethernet0/0, from which the VNET sub-interfaces will receive their derived configuration. Configuration of some non-default values such as hello-interval, hold-time and authentication has been done to demonstrate the inheritance. You will also notice the VNET sub-mode under af-interface in the global EIGRP process. This is a way to control which configuration options are cloned to the dynamically created af-interface for each VNET within its EIGRP vrf configuration.

For example the derived config for Eth0/0 in the global routing table is inherited from vnet global (hello-interval 30, hold-time 90). The authentication-mode hmac-sha-256 for Eth0/0 is configured directly on this af-interface in the running-config, and the derived config output shows that Eth0/0

has inherited the command. Since authentication mode is configured on the VNET trunk af-interface, it is inherited by all VNET interfaces.

For vrf orange, VNET orange has been configured with a hello-interval of 15 in the running-config. In the derived config you can see for VRF orange in autonomous-system 101, the hello-interval of 15 was taken from the VNET sub-mode under af-interface eth0/0, in the global process. The hold-time was not modified and was cloned from af-interface eth0/0 which is using the default value.

VNET red has no configuration differences from af-interface Eth0/0, so it inherits default timer values as well as the authentication mode.

These configuration options allow flexibility for the operator to use different parameters for each VNET trunk sub-interface. For example, different timer values, authentication modes or passive interface. To summarize the inheritance rules, all VNETs will inherit the configuration from the VNET trunk af-interface. The VNET specific configuration in VNET sub-mode will also be inherited by the VNET trunk sub-interfaces, and takes priority over the parameters from the af-interface.

Below is some additional output to verify the configuration inheritance:

```
R1#show eigrp address-family ipv4 interface detail e0/0
EIGRP-IPv4 VR(named) Address-Family Interfaces for AS(100)
Xmit Queue PeerQ Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable Un/Reliable SRTT Un/Reliable Flow Timer Routes
Et0/0 1 0/0 0/0 6 0/2 50 0
Hello-interval is 30, Hold-time is 90
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 3/1
Hello's sent/expedited: 2959/3
Un/reliable mcasts: 0/4 Un/reliable ucasts: 5/5
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 3 Out-of-sequence rcvd: 1
Topology-ids on interface - 0
Authentication mode is HMAC-SHA-256, key-chain is not set
Topologies advertised on this interface: base
Topologies not advertised on this interface:
```

```
R1#show eigrp address-family ipv4 vrf orange interface detail e0/0.101
EIGRP-IPv4 VR(named) Address-Family Interfaces for AS(101)
VRF(orange)
Xmit Queue PeerQ Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable Un/Reliable SRTT Un/Reliable Flow Timer Routes
Et0/0.101 1 0/0 0/0 5 0/2 50 0
Hello-interval is 15, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 4/1
Hello's sent/expedited: 2371/3
Un/reliable mcasts: 0/4 Un/reliable ucasts: 6/5
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 3 Out-of-sequence rcvd: 1
Topology-ids on interface - 0
Authentication mode is HMAC-SHA-256, key-chain is not set
Topologies advertised on this interface: base
Topologies not advertised on this interface:
```

```
R1#show eigrp address-family ipv4 vrf red interface detail e0/0.102
EIGRP-IPv4 VR(named) Address-Family Interfaces for AS(102)
VRF(red)
Xmit Queue PeerQ Mean Pacing Time Multicast Pending
```

```

Interface Peers Un/Reliable Un/Reliable SRTT Un/Reliable Flow Timer Routes
Et0/0.102 1 0/0 0/0 4 0/2 50 0
Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 6/1
Hello's sent/expedited: 2676/3
Un/reliable mcasts: 0/6 Un/reliable ucasts: 7/5
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 3 Out-of-sequence rcvd: 1
Topology-ids on interface - 0
Authentication mode is HMAC-SHA-256, key-chain is not set
Topologies advertised on this interface: base
Topologies not advertised on this interface:

```

Route Replication with EIGRP name mode

One of the benefits of EVN is the ability to replicate routes between VNETs. For example R4 in VRF red may need to reach a service on 192.168.13.0/24 which is part of VRF orange. This can be achieved using the configuration below.

```

R2#show run
vrf definition orange
vnet tag 101
!
address-family ipv4
exit-address-family
!
vrf definition red
vnet tag 102
!
address-family ipv4
route-replicate from vrf orange unicast eigrp 101 route-map filter
exit-address-family
!
<output removed>
!
ip prefix-list filter seq 5 permit 192.168.13.0/24
!
route-map filter permit 10
  match ip address prefix-list filter
!

```

Now 192.168.13.0/24 prefix is in VRF red, however the ping is not working because source address is not route replicated into VNET orange.

```

R2#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
a - application route
+ - replicated route, % - next hop override, p - overrides from Pfr

```

```

Gateway of last resort is not set

```



```

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D 10.5.5.5/32 [90/1536640] via 10.12.12.1, 03:48:46, Ethernet0/0.102
D 10.6.6.6/32 [90/1024640] via 192.168.26.6, 03:48:37, Ethernet2/0
C 10.12.12.0/24 is directly connected, Ethernet0/0.102
L 10.12.12.2/32 is directly connected, Ethernet0/0.102
D + 192.168.13.0/24
[90/1536000] via 10.12.12.1 (orange), 03:48:46, Ethernet0/0.101
D 192.168.15.0/24 [90/1536000] via 10.12.12.1, 03:48:46, Ethernet0/0.102
192.168.26.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.26.0/24 is directly connected, Ethernet2/0
L 192.168.26.2/32 is directly connected, Ethernet2/0
R2#
R2#
R2#ping vrf red 192.168.13.1 source e2/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.13.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.26.2
.....
Success rate is 0 percent (0/5)

```

After all replicated routes from VRF red to VRF orange on R1, using similar configuration:

```
R2#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
a - application route
+ - replicated route, % - next hop override, p - overrides from PFR

```

```
Gateway of last resort is not set
```

```

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D 10.5.5.5/32 [90/1536640] via 10.12.12.1, 03:48:46, Ethernet0/0.102
D 10.6.6.6/32 [90/1024640] via 192.168.26.6, 03:48:37, Ethernet2/0
C 10.12.12.0/24 is directly connected, Ethernet0/0.102
L 10.12.12.2/32 is directly connected, Ethernet0/0.102
D + 192.168.13.0/24
[90/1536000] via 10.12.12.1 (orange), 03:48:46, Ethernet0/0.101
D 192.168.15.0/24 [90/1536000] via 10.12.12.1, 03:48:46, Ethernet0/0.102
192.168.26.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.26.0/24 is directly connected, Ethernet2/0
L 192.168.26.2/32 is directly connected, Ethernet2/0
R2#
R2#
R2#ping vrf red 192.168.13.1 source e2/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.13.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.26.2
.....
Success rate is 0 percent (0/5)

```

Note: You can route-replicate connected, BGP, EIGRP, etc. Please refer to the references for more examples.

Routing Context

Another nice feature with EVN is the concept of routing context. This allows you to execute commands within VRF red, without having to include 'vrf red' into each CLI. For example, the same ping as above using routing context is shown below.

```
R2#routing-context vrf red
R2%red#ping 192.168.13.1 source e2/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.13.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.26.2
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R2%red#
```

Enhanced traceroute

The output of the traceroute command will also display VNET VRF names, which is helpful for troubleshooting, especially if route replication is involved.

```
R6#traceroute 192.168.13.3
Type escape sequence to abort.
Tracing the route to 192.168.13.3
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.26.2 (red,orange/101) 1 msec 0 msec 0 msec
 2 10.12.12.1 (orange/101,orange) 2 msec 1 msec 1 msec
 3 192.168.13.3 0 msec * 1 msec
```

The same trace from R2

```
R2#trace vrf red 192.168.13.3 source 192.168.26.2
Type escape sequence to abort.
Tracing the route to 192.168.13.3
VRF info: (vrf in name/id, vrf out name/id)
 1 10.12.12.1 (orange/101,orange) 1 msec 1 msec 0 msec
 2 192.168.13.3 1 msec * 1 msec
```

In this output you can see that from R2, the next-hop in VRF orange is taken directly to reach 192.168.13.0/24.

Conclusion

EVN VNET configuration with EIGRP named mode provides a way for customers to deploy a virtualized network environment, and remove some of the complexity associated with traditional MPLS VPN, or VRF-lite. Understanding the inheritance rules is key to successfully deploying this feature and ensuring the network is operating as intended.

References

Easy Virtual Networks whitepaper

http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/layer-3-vpns-l3vpn/whitepaper_c11-638769.html

Configuration guide

<http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/evn/configuration/xe-3s/evn-xe-3s-book/evn-overview.html>