

Configurez le réseau virtuel facile avec Mode nommé par EIGRP

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Configurez](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérifiez](#)

[Inheritance avec le mode Désigné EIGRP](#)

[Réplication d'artère avec le mode de nom EIGRP](#)

[Acheminement du contexte](#)

[Traceroute améliorée](#)

[Conclusion](#)

[Références](#)

Introduction

Le but de ce document est d'expliquer la configuration d'EVN (réseau virtuel facile) utilisant le mode Désigné EIGRP (Enhanced Interior Gateway Routing Protocol). C'est un supplément au document [virtuel facile de configuration réseau](#), qui explique l'utilisation d'OSPF (protocole OSPF), aussi bien qu'à d'autres sujets avancés comme les listes de joncteur réseau VNET et la réplication d'artère. L'EVN VNET a été destiné pour que des opérateurs aient une option plus facile que MPLS (commutation par étiquette multi de Protocol) VPN (réseau privé virtuel) ou Vrf-lite (Virtual Routing and Forwarding) pour déployer le plusieurs vrf. EVN VNET emploie un concept de configuration copiée pour que l'interface de protocoles de routage et de joncteur réseau VNET retire la charge de l'opérateur et pour sauvegarde certaines des tâches répétitives. Le dépannage de l'EIGRP, routage ou CEF (Cisco Express Forwarding) est en dehors de la portée de ce document et à moins que remarquable vous puissiez suivre des procédures de dépannage normales.

Conditions préalables

Conditions requises

Cisco recommande que vous ayez la connaissance de base de l'EIGRP.

Cette caractéristique est avaiable dans peu de versions après la version IOS 15.2. Pour vérifier si le mode nommé par EIGRP avec EVN VNET est pris en charge, vérifiez la sortie des **modules d'extension de show ip eigrp**. Si la version 1.00.00 ou ultérieures facile de réseau virtuel est présente, alors votre version prend en charge cette caractéristique.

```
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
```

Remarque: Le mode nommé par EIGRP avec EVN VNETs n'est pas pris en charge dans 15.1SY. Dans cette version vous devez utiliser la configuration classique du mode EIGRP qui est déjà expliquée dans la documentation disponible.

Le BFD (détection bidirectionnelle d'expédition) est actuellement seulement pris en charge sur VNET global et ne fonctionnera sur aucune sous-interface Désignée VNET sur le joncteur réseau VNET.

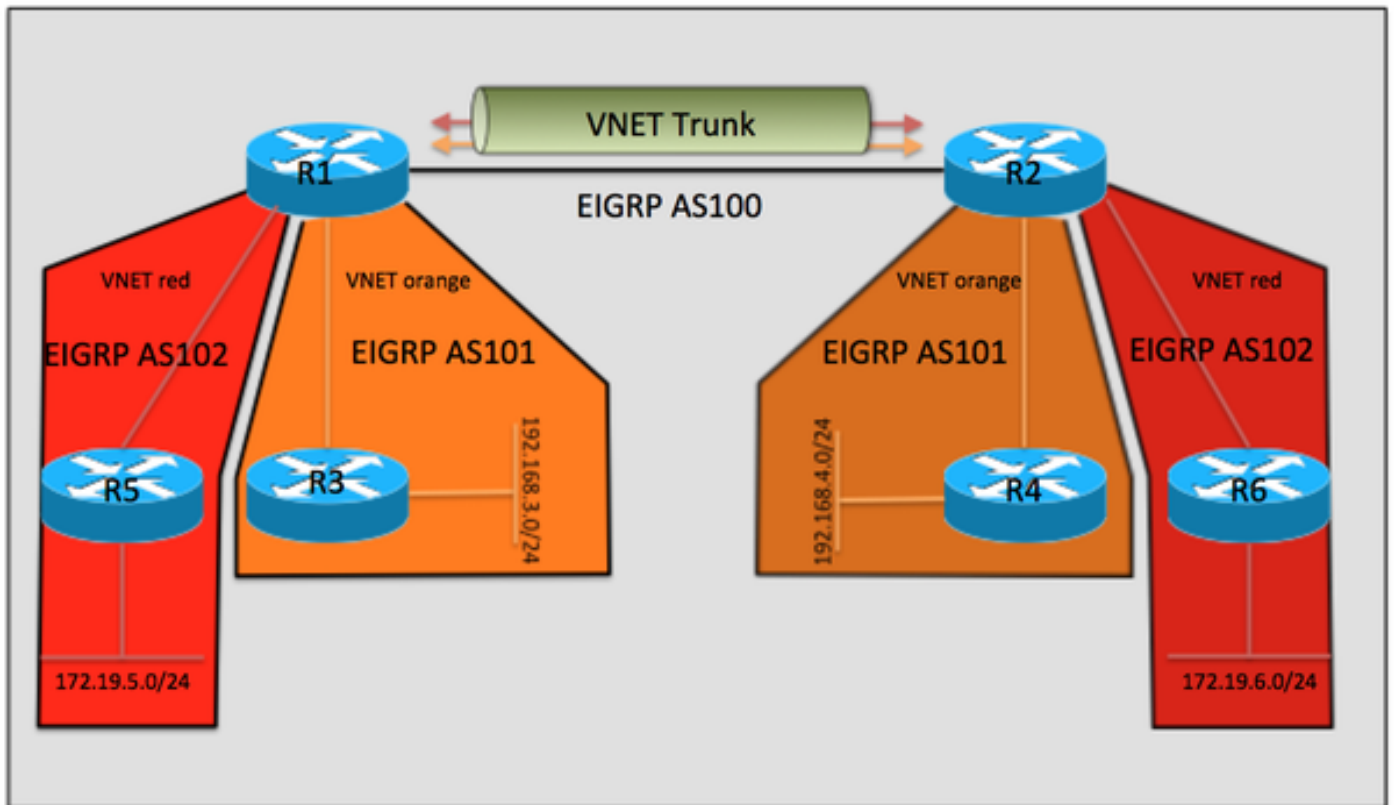
On ne lui informe pas utiliser le par défaut d'af-interface en utilisant l'EIGRP a nommé le mode avec EVN VNETs dû à l'héritage imprévisible possible.

[Composants utilisés](#)

Les informations dans ce document ont été créées des périphériques dans une version 15.6(1)S2 courante de Cisco IOS d'environnement de travaux pratiques de particularité. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Configurez

[Diagramme du réseau](#)



Configurations

Les configurations de R3, de R4, de R5 et de R6 sont tous semblables, et donc parti hors du document. Ils sont simplement configurés pour former un voisin EIGRP avec R1 ou R2, et ils ne se rendent pas compte de l'EVN VNET utilisé entre R1 et R2.

Configuration appropriée de 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

```

Configuration appropriée de 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

```

Vérifiez

Un des avantages du réseau virtuel facile est la simplicité de la configuration. Ceci est réalisé en configurant automatiquement les joncteurs réseau VNET pour chaque balise VNET. Comparant EVN à Vrf-lite, chaque sous-interface devrait être manuellement configurée. Ethernet0/0 est le joncteur réseau VNET qui connecte R1 et R2, et une sous-interface VNET est automatiquement créée pour que chaque VNET réponde aux exigences de séparation du trafic pour EVN en ajoutant des trames avec une balise dot1Q VNET. Ces sous-interfaces ne sont pas visibles dans la sortie du show running-configuration, toutefois elles peuvent être vues avec le 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

```

De même, vous pouvez voir que la configuration EIGRP est également créée automatiquement :

```

R1#show derived-config | sec router eigrp
router eigrp named
!

```

```

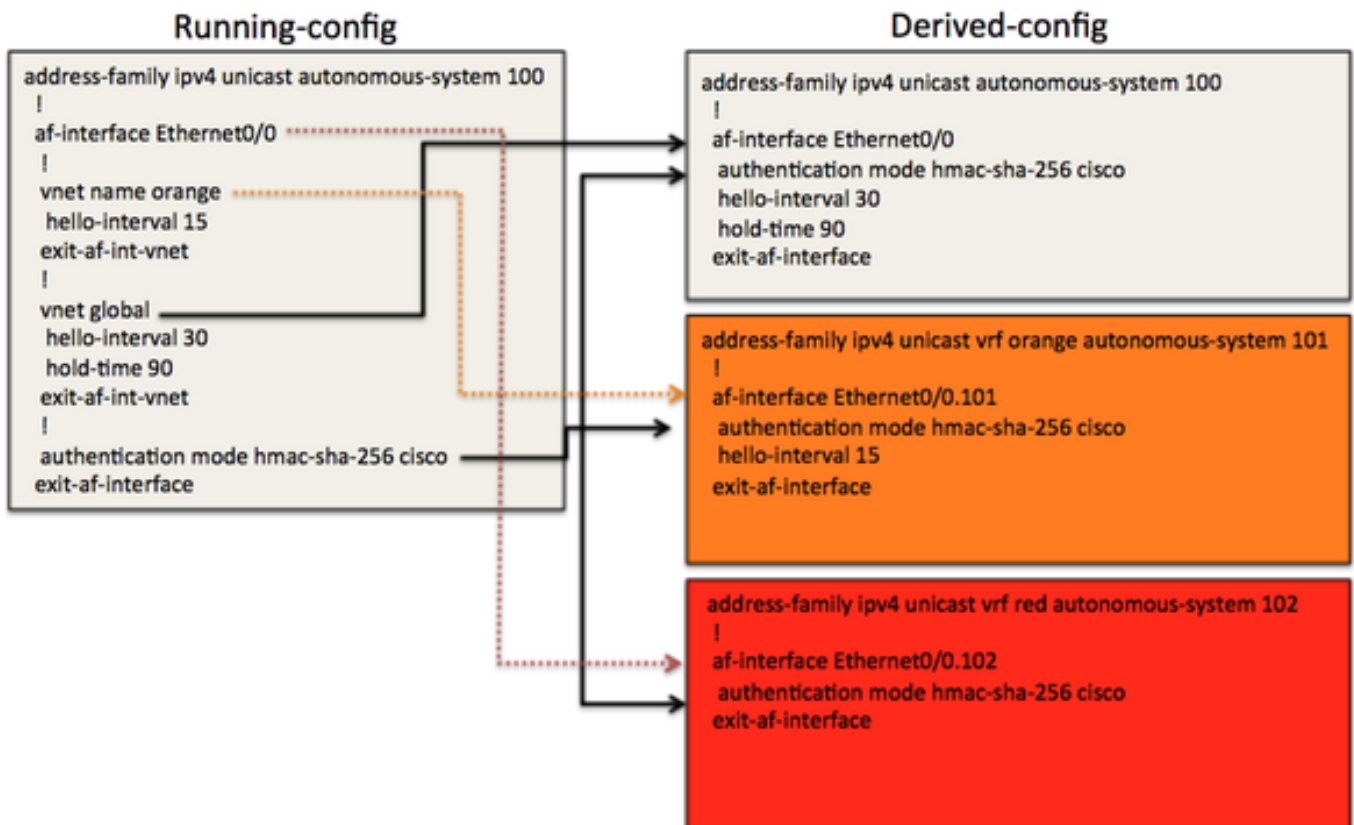
address-family ipv4 unicast autonomous-system 100
!
af-interface Ethernet0/0
authentication mode hmac-sha-256 cisco
exit-af-interface
!
topology base
exit-af-topology
network 10.0.0.0
exit-address-family
!
  address-family ipv4 unicast vrf orange autonomous-system 101
!
  af-interface Ethernet0/0.101
  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#

```

Une observation intéressante dans la sortie ci-dessus est l'héritage d'af-interface pour les sous-interfaces VNET de l'af-interface ethernet0/0 dans l'autonomous-system global 100 de vrf. La section suivante explique ceci plus en détail :

Inheritance avec le mode Désigné EIGRP

La figure ci-dessous sera utilisée pour aider à visualiser les règles d'héritage quand utilisant l'EIGRP a nommé le mode avec EVN VNETs.



Dans l'exemple ci-dessus il y a une af-interface ethernet0/0 de joncteur réseau VNET, dont les sous-interfaces VNET recevront leur configuration dérivée. La configuration de certaines valeurs autres que par défaut telles que l'intervalle Hello, le temps de maintien et l'authentification a été faite pour expliquer l'héritage. Vous noterez également le sous-modèle VNET sous l'af-interface dans le processus global EIGRP. C'est une manière de contrôler que des options de configuration sont copiées à l'af-interface dynamiquement créée pour chaque VNET dans sa configuration de vrf EIGRP.

Par exemple le config dérivé pour Eth0/0 dans la table globale de routage est hérité du vnet global (intervalle Hello 30, temps de maintien 90). L'authentification mode hmac-sha-256 pour Eth0/0 est configurée directement sur cette af-interface dans le running-config, et la sortie dérivée de config prouve qu'Eth0/0 a hérité de la commande. Puisque l'authentification mode est configurée sur l'af-interface de joncteur réseau VNET, elle est héritée par toutes les interfaces VNET.

Pour l'orange de vrf, l'orange VNET a été configurée avec un intervalle Hello de 15 dans le running-config. Dans le config dérivé que vous pouvez voir pour l'orange de VRF dans l'autonomous-system 101, l'intervalle Hello de 15 a été pris du sous-modèle VNET sous l'af-interface eth0/0, dans le processus global. Le temps de maintien n'a pas été modifié et a été copié de l'af-interface eth0/0 qui utilise la valeur par défaut.

Le rouge VNET n'a aucune différence de configuration de l'af-interface Eth0/0, ainsi il hérite des valeurs de minuteur par défaut aussi bien que de l'authentification mode.

Ces options de configuration permettent la flexibilité pour que l'opérateur utilise différents paramètres pour chaque sous-interface de jonction VNET. Par exemple, différentes valeurs de temporisateur, authentifications mode ou interface passive. Pour récapituler les règles d'héritage, tout les VNETs hériteront de la configuration de l'af-interface de joncteur réseau VNET. La configuration spécifique VNET dans le sous-modèle VNET sera également héritée par les sous-interfaces de jonction VNET, et prend la priorité au-dessus des paramètres de l'af-interface.

Est ci-dessous une certaine sortie supplémentaire pour vérifier l'héritage de configuration :

```
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:
```

Réplication d'artère avec le mode de nom EIGRP

Un des avantages d'EVN est la capacité de répliquer des artères entre VNETs. Par exemple R4

en rouge de VRF peut devoir atteindre un service sur 192.168.13.0/24 qui fait partie d'orange de VRF. Ceci peut être réalisé utilisant la configuration ci-dessous.

```
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
!
```

Maintenant le préfixe 192.168.13.0/24 est en rouge de VRF, toutefois le ping ne fonctionne pas parce que l'adresse source n'est pas artère répliquée dans l'orange VNET.

```
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)
```

Après tout artères répliquées du rouge de VRF à l'orange de VRF sur R1, utilisant la configuration semblable :

```
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)
```

Remarque: Vous pouvez route-replicate connecté, BGP, EIGRP, etc. Veuillez se référer aux références pour plus d'exemples.

Acheminement du contexte

Une autre fonctionnalité intéressante avec EVN est le concept du contexte de routage. Ceci te permet pour exécuter des commandes dans le rouge de VRF, sans devoir inclure le « rouge de vrf » dans chaque CLI. Par exemple, les mêmes ping comme ci-dessus utilisant le contexte de routage est affiché ci-dessous.

```
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#
```

Traceroute améliorée

La sortie de la commande traceroute affichera également des noms de VRF VNET, qui est utile pour le dépannage, particulièrement si la réplication d'artère est impliquée.

```
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
```

Les mêmes suivi de 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
```

Dans cette sortie vous pouvez voir que de R2, l'orange de VRF de next-hop in est prise directement pour atteindre 192.168.13.0/24.

Conclusion

La configuration EVN VNET avec le mode nommé par EIGRP fournit une manière pour que les clients déploient un environnement de réseau virtualisé, et enlève une partie de la complexité associée avec MPLS traditionnel VPN, ou Vrf-lite. La compréhension des règles d'héritage est principale à déployer avec succès cette caractéristique et assurer le réseau fonctionne comme prévu.

Références

Whitepaper facile de réseaux virtuels

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

Guide de configuration

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