\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Routing with VRF

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Author*:

Gabriel Rosas



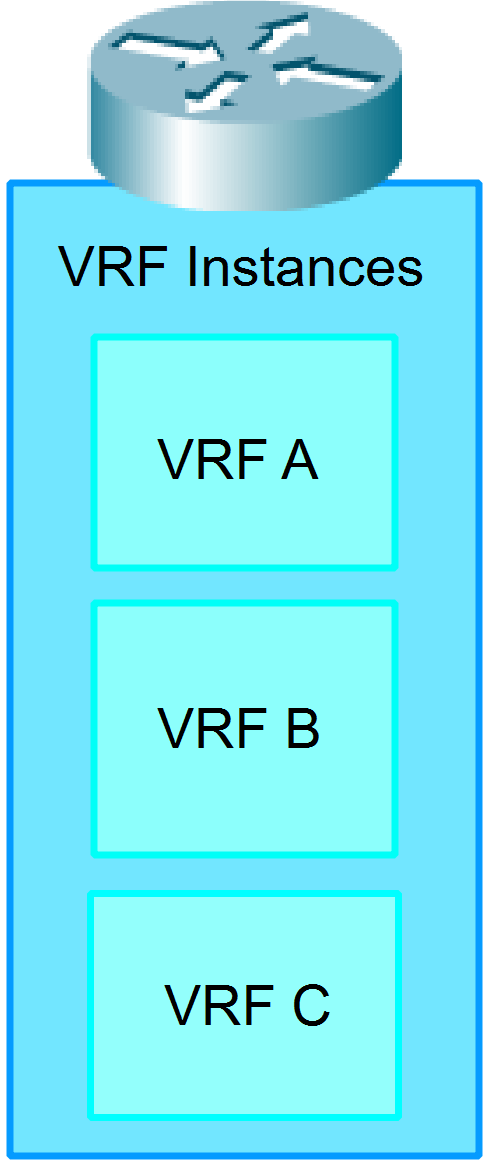
Purpose

Do you hate sharing? Want your packets to travel privately, isolated from other traffic? *Virtual Routing and Forwarding* (VRF) is perfect for you! In this lab, I configure VRF across four routers, splitting these routers into two VRFs: one that routes using EIGRP and one that routes using BGP. Packets routed on the EIGRP network are separated from those routed on the BGP network.

Background Information

What is VRF?

Have you ever thought to run Linux on Windows? It is possible through *virtualization*, which enables one host to run multiple “guest” operating systems. *Virtual Routing and Forwarding* provides the similar features, but for routers. Suppose I want a router to route three different types of traffic; I could create three VRF instances, *VRF A*, *VRF B*, and *VRF C*:



One VRF may route the traffic of a department, another for public use, and the last for network admins. We don’t want these groups interacting, so we can split the router into multiple instances to route each group separately. Unlike VLANs, a hacker cannot “hop” and gain access to a different VRF, making them more secure and protected. Each VRF instance is dedicated to routing, and as such, a unique routing table is created for each VRF. But what is a routing table?

Routing tables

Like a signpost at the edge of a crossroads, routers contain a list of directions for different destinations. A packet arrives at a router. You can think of a router like an *intersection* or *crossroads*, interfaces representing a different path to take. This router has three interfaces: north, south, and east. The packet arrived on the east interface, so it either must turn north or south, assuming one of these paths lead to the destination. Luckily, there are directions in the router: *10.0.0.0/24* out interface *north*; *172.16.0.0/24* out interface *south*. The packet has a destination address of *10.0.0.3*, which matches up with the *north* interface. The router sends the packet out the north interface. Routes are either generated statically, by the admin, or automatically by routing protocols such as OSPF, BGP, etc.

Here is an example of a routing table:

|  |
| --- |
| Gateway of last resort is not set  10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks  O IA 10.10.10.0/30 [110/128] via 10.10.10.5, 01:03:27, Serial0/1/1  C 10.10.10.4/30 is directly connected, Serial0/1/1  L 10.10.10.6/32 is directly connected, Serial0/1/1  C 10.10.10.8/30 is directly connected, Serial0/1/0  L 10.10.10.9/32 is directly connected, Serial0/1/0  O IA 10.10.10.12/30 [110/128] via 10.10.10.10, 01:03:27, Serial0/1/0  C 10.10.10.16/30 is directly connected, Serial0/2/0  L 10.10.10.17/32 is directly connected, Serial0/2/0  O IA 10.10.10.20/30 [110/128] via 10.10.10.18, 01:03:27, Serial0/2/0  O IA 10.10.10.24/30 [110/192] via 10.10.10.18, 01:03:27, Serial0/2/0  O E2 10.10.10.28/30 [110/100] via 10.10.10.18, 01:03:27, Serial0/2/0 |

Ignoring the letters on the left (the origin of the route), we can see a range of accessible addresses and the corresponding interface leading towards them. For example, *10.10.10.0/30* addresses direct out the *Serial0/1/1* interface. *Via \*ip\**, is also commonly seen as a direction, indicating that a packet should be sent to the specified neighboring router. Sometimes there is a combination of directions: both *interface* and *neighboring ips*.

VRF Route Distinguishers

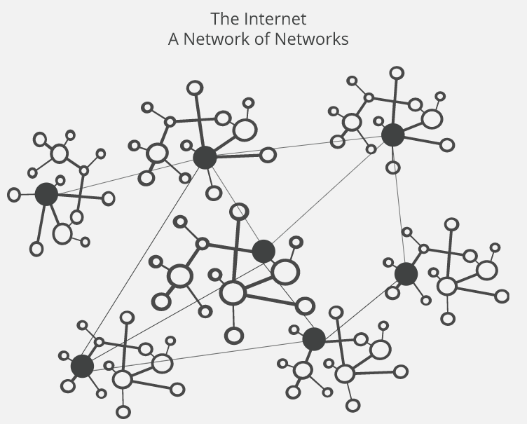
*Route distinguishers* (RD) allow routers to keep track of what routes belong to what VRF. Like the name implies, they *distinguish* one set of routes from another. There are different formats of RD:

1. Type 0: 2-byte ASN; 4-byte value
2. Type 1: 4-byte IP; 2-byte value
3. Type 2: 4-byte ASN; 2-byte value

For example, *100:4200000000*, *10.0.0.1:200*, *30000000:300*, are all valid RDs in the various formats, all solely cosmetic. VRF routes are identified in the form *RD:IP*, for example, *65000:1:10.0.0.0/24*.

What is BGP?

*BGP* (Border Gateway Protocol) is the most popular routing protocol, commonly used by ISPs (Internet Service Providers) to route customer traffic. Without BGP, the internet would not function nearly as well, if at all. Think of BGP as the postal service that delivers a letter to the recipient in the fastest and most efficient manner possible. When someone submits data across the internet, BGP is responsible for choosing the best path out of all preexisting available paths, which usually means passing through autonomous systems.



*An example of Autonomous systems and their local networks*

So, what are autonomous systems? Autonomous systems (AS) are a collection of routers, each with their own lesser hierarchy of routers that eventually connects to local networks. Each autonomous system is aware of other autonomous system(s) and can broadly determine where to route traffic based on which autonomous system holds the desired destination. ASes typically belong to ISPs (Internet service providers) or other large high-tech organizations, such as tech companies, universities, or scientific institutions. The internet is run under a collection of autonomous systems.

Kingdom Analogy

I suppose one could think of an autonomous system as a form of kingdom. Each kingdom has a ruler that dictates certain policies that the underlying citizens and infrastructure abide to. For example, if a kingdom is landlocked, it likely has a high demand for fish and salt. Therefore, a *policy* is implemented where all traders from the nearest port town have free access to and from the kingdom. Different autonomous systems often have these unique routing *policies*.

There are many paths and roads in the kingdom internally, so much so that if one goes down, alternate routes are readily available. Some kingdoms have routes bridging them, but often a traveler (packet) will have to journey through multiple kingdoms to reach their desired destination. In other words, a packet may have to pass through multiple ASes to reach its destination.

Each AS is assigned a unique, 32-bit number, the *Autonomous System Number* (ASN). These numbers differentiate what “kingdom” a router falls under. Routers with the same AS are part of the same kingdom. To qualify for an ASN, one needs proof of a unique routing policy, knowledge on how to link autonomous systems, and a plentiful quantity of hosts. There is no point in creating a kingdom with only a handful of hosts. If you satisfy these rules, then the closest *regional internet registry* (RIR), may delegate you an ASN.

Internal and External BGP

As I vaguely covered in my analogy, there are two types of BGP: *internal BGP* (within kingdoms) and *external BGP* (between kingdoms).

*External BGP* (eBGP) is the bridge that connects autonomous systems, where neighbors can broadly exchange network prefixes to learn more about each other’s networks.

*Internal BGP* (iBGP) is a TCP based protocol to help advertise and support eBGP routes. The kicker: iBGP alone does not do any routing. To route, one needs an IP based protocol. So why bother with iBGP at all?

Consider an old, flimsy wooden bridge. Driving a cargo truck across would collapse the bridge. But now, with iBGP, that bridge is reinforced with a concrete foundation, metal bearings, and arches to brace the heavy loads. BGP is the only protocol designed to support the hundreds of thousands of routes that make up the internet. As of writing this, the size of the full IPv4 BGP routing table is around 800,000 prefixes without even accounting for IPv6. For reference, the average OSPF router would suffer at around 6000 prefixes. This is why iBGP is oftenly used in conjunction with an IGP; the IGP does the local routing whilst iBGP contains the major routing table.

Both internal and external BGP sessions establish neighbors based on a peering system. You define a peer with a neighbor statement: for example, *neighbor 10.0.0.1 remote-as 100* states that there is a router connected, *10.0.0.1* running under ASN *100*. The neighbor *10.0.0.1* would need to define this router as a neighbor for a complete peer adjacency to form. Once both routers point to each other, they are peered. Networks are advertised with network statements: for example, *network 10.0.0.0 mask 255.255.255.0* will add the prefix *10.0.0.0/24* to the routing table. Other routers will direct traffic for *10.0.0.0/24* towards the router with the network statement.

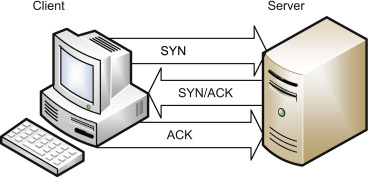
How does BGP function?

The main purpose BGP serves is forwarding traffic to an external network in the most efficient manner possible. Some factors that determine the best path are:

1. The path with the highest *weight*. This is a user defined variable.
2. The path with the highest*LOCAL\_PREF*. Local preference determines which path is preferred when leaving a local AS.
3. The path with the highest*AS\_PATH*. The main purpose of *AS\_PATH* is to prevent infinite routing table updates. It is rather complicated, but essentially if a router goes down in a network, then this might cause the other routers to falsely change their paths, resulting in an infinite loop of changing paths. This can only happen in a distance-vector routing protocol such as BGP or RIP.
4. Favoring *eBGP* paths over iBGP paths.

BGP is a *distance-vector* routing protocol. Distance-vector routing protocols work by advertising routing tables to their neighbors. If the routes from the neighbor are better than the ones they currently have, the router will update its routing table to the preferable routes. Like all other routing protocols, BGP must first establish a neighbor adjacency with another BGP router to be able to exchange routing information. Unlike other routing protocols, BGP does not broadcast or multicast to discover other BGP neighbors. Neighbor relationships must be established manually and BGP uses TCP port 179 for the connection. There are a couple of different states BGP routers may encounter when becoming neighbors:

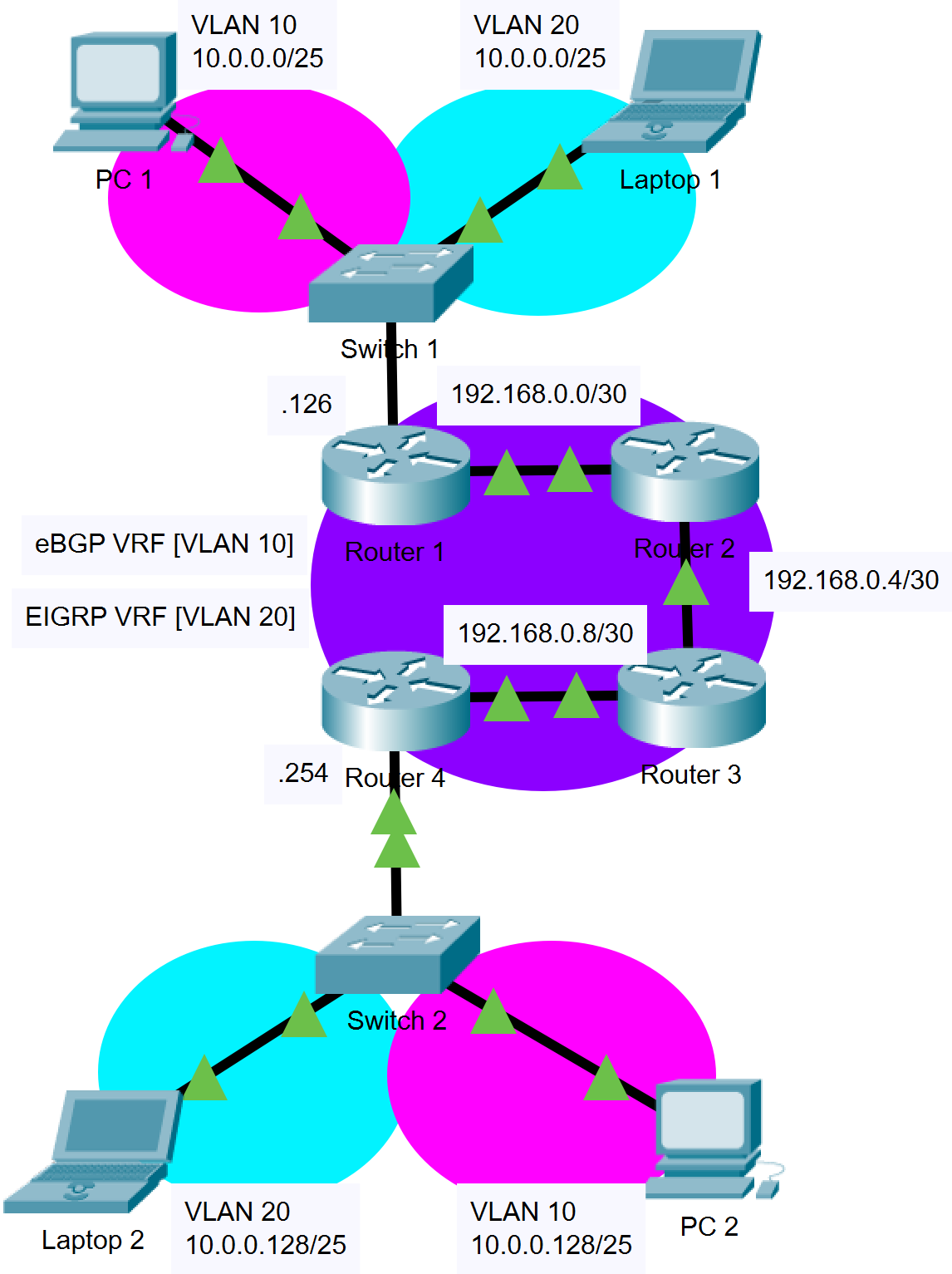
1. Idle. In Idle, BGP waits for a “start event”. This could be when a new BGP neighbor is configured or when a reset occurs between peers that already had a connection. After the start event, BGP will initialize a TCP connection with the remote neighbor and initialize some functions. In success, BGP moves to the *Connect* state, while in failure, BGP remains in the *Idle* state.
2. Connect. In *Connect*, BGP waits for the TCP three-way handshake to complete. Both sides need to *synchronize* (SYN) and *acknowledge* (ACK) each other in a TCP three-way handshake. If the results are successful, BGP advances to the *OpenSent* state. If the results are unsuccessful, BGP begins the *Active* state.



1. Active. BGP will try another TCP three-way handshake to establish a connection with the remote neighbor. On success, BGP will transition to the *OpenSent* state. After a certain amount of time has passed with no success, BGP will revert to the *Connect* state.
2. OpenSent. BGP will wait for an “open message” from the remote neighbor. Open messages contain information about the BGP router, such as version, ASN, BGP router ID, and hold time. If the versions or hold times mismatch, BGP reverts to the *Idle* state. The ASN determines whether the BGP session will be running iBGP or eBGP. If the TCP session ever fails, BGP will revert to the *Active* state. If all passes, BGP will start sending keepalive messages to maintain the TCP session.
3. OpenConfirm. BGP waits for a keepalive message from the remote BGP neighbor. When keepalive messages are consistently received, BGP moves to the *Established* state. In any other case, BGP falls back to the *Idle* state.
4. Established. The neighbor adjacency has been formed. As long as keepalive messages are sent, the neighborship remains up. Otherwise, BGP resets back to Idle state.

Adjacencies are often formed by defining the *directly connected* interface as a neighbor, a common trait in most routing protocols. However, a technique when working with BGP is to use loopback interfaces as neighbors. Using loopbacks is common for iBGP but it also works with eBGP. Loopbacks are preferred because of redundancy: if the physical interface goes down, perhaps due to hardware, loopback interfaces will stay up since they are *virtual*.

Network Diagram



Process

I began my lab with some brief research on the concepts of VRF. All example configurations I found were very similar to regular routing, with the occasional “*vrf*” statement embedded in the typical command. VRF was beginning to seem trivial – the real complications would be the routing protocols themselves. But before I could implement routing, I needed a clear topology with proper addressing.

The goal was to have two VRF networks, incorporating four routers, each VRF routed through any protocol we chose. I decided to route using *BGP* and *EIGRP*. In my topology, I created two VLANs, 10 and 20, which my *BGP* and *EIGRP* VRFs would route, respectively. Addressing came next: for point-to-point networks (router to router), I like using [/30] subnets, allocating me two addresses, perfect for a network with two devices; for the end users in the VLANs, I chose [/25] subnets so each VLAN pair would end up as a clean *10.0.0.0/24* network.

With the topology complete, I could finally configure the routers. On each router, I set up interfaces with IPs according to the topology and created two VRFs, creatively named *BGP* and *EIGRP*. You can imagine which VRF was assigned what routing protocol. Once the green lights started blinking on the router, indicating the interfaces were up, I was ready to get my first VRF working: *BGP*.

Configuring a BGP VRF was much like “normal” BGP, which I have done in the past. The main difference was the new concept of *route distinguishers*, with the introduction of VRF. BGP requires a route distinguisher when creating a routing instance, something I learnt firsthand after attempting to enter the *address-family*. “*VRF BGP does not have an RD configured*” would pop up each time. I fixed this problem by adding “*rd 10:1*” in the BGP VRF.

Lastly was EIGRP. Unlike BGP, EIGRP did not require a defined *route distinguisher*. My partner and I set up the EIGRP network, but nothing was appearing in the routing table. We tried assigning *route distinguishers*, checked *IPs*, and referenced sources online. Everything appeared correct. A while later, after double checking for the fifth time, I noticed the *neighbor statement* might be wrong. Since we were using sub-interfaces, EIGRP’s neighbor statements looked almost identical: for example, *GigabitEthernet0/0/0.10* and *GigabitEthernet0/0/0.20* both point out *GigabitEthernet0/0/0* but are tagged differently. Surely enough, this slight difference was the problem.

Lab Commands

|  |  |
| --- | --- |
| **Command** | A statement necessary for a configuration to work, denoted in bold |
| **[*Argument*]** | An argument necessary for a command to function, denoted in bold italics. |
| *Optional-Statement*  *<Optional Argument>* | An optional argument or statement, not necessary for a command to function, denoted in italics |

Router(config)# **interface [*interface*] [*id*]**

* Enables configuration on a specific interface.

// VRF

Router(config)# **ip vrf [*name*]**

* Creates a vrf

*VRFs are used to instance routing tables.*

Router(config-vrf)# **rd [*asn*]:[*arbitrary number*]**

* Creates a route distinguisher

*The route distinguisher is like an* id *for a VRF. Each routing instance is grouped based on the rd prefix at the beginning of routes in the table.*

Router(config-subif)# **ip vrf forwarding [*name*]**

* Assigns a *sub-interface* to route traffic of a specified vrf

*This command is an interface command, typed in* interface *or* sub-interface *mode.*

// BGP

Router(config)# **router bgp [*autonomous system number*]**

* Activates a BGP router and enters router configuration mode

*The autonomous system number (ASN) is a number that identifies a large collection of routers on the internet. Typically, there are networks run under an ASN by a technical administration. eBGP connects different autonomous systems while iBGP is run within each ASN.*

Router(config-router)# **address-family [*protocol*]** *vrf <vrf name>*

* Enters configuration mode for a BGP address family

*I like to think of address-families as workspaces for certain IP protocols. For example, IPv6 is configured in the IPv6 address family. This is where redistribution, network statements or activation commands occur. Address-families can be implemented for separate VRFs by adding “vrf VRF”.*

Router(config-router-af)# **neighbor [*ip address*] remote-as [*neighbor’s asn*]**

* Used in forming BGP neighbor adjacencies.

*This command takes an IP address of a neighbor’s interface and the ASN of the neighbor. For a BGP neighborship to be established, each router must have* routes to the neighbor’s IP *and the* correct IP and ASN of their neighbor*.*

Router(config-router-af)# **network [*network address*] mask [*subnet mask*]**

* Advertises a directly connected network to the BGP routing table

*BGP’s network statements are not to be confused with OSPF or EIGRPs; they aren’t used to form adjacencies between BGP routers. A BGP network statement is configured alongside a neighbor statement, the former advertising the network and the latter the neighbor establishment.*

Configurations

Router 1

R1#show running-config

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R1

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf BGP

rd 10:1

ip vrf EIGRP

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214421CF

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.1 255.255.255.252

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.1 255.255.255.252

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 10.0.0.126 255.255.255.128

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 10.0.0.126 255.255.255.128

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router eigrp 20

address-family ipv4 vrf EIGRP

network 10.0.0.0

network 192.168.0.0

neighbor 192.168.0.2 GigabitEthernet0/0/0.20

autonomous-system 20

eigrp router-id 1.1.1.1

exit-address-family

router bgp 10

bgp router-id 1.1.1.1

bgp log-neighbor-changes

address-family ipv4 vrf BGP

network 10.0.0.0 mask 255.255.255.128

neighbor 192.168.0.2 remote-as 20

neighbor 192.168.0.2 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R1#sh ip route vrf BGP

Routing Table: BGP

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, 3 subnets, 2 masks

C 10.0.0.0/25 is directly connected, GigabitEthernet0/0/1.10

L 10.0.0.126/32 is directly connected, GigabitEthernet0/0/1.10

B 10.0.0.128/25 [20/0] via 192.168.0.2, 00:04:20

192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.0.0/30 is directly connected, GigabitEthernet0/0/0.10

L 192.168.0.1/32 is directly connected, GigabitEthernet0/0/0.10

R1#sh ip route vrf EIGRP

Routing Table: EIGRP

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, 3 subnets, 2 masks

C 10.0.0.0/25 is directly connected, GigabitEthernet0/0/1.20

L 10.0.0.126/32 is directly connected, GigabitEthernet0/0/1.20

D 10.0.0.128/25

[90/28928] via 192.168.0.2, 00:02:15, GigabitEthernet0/0/0.20

192.168.0.0/24 is variably subnetted, 4 subnets, 2 masks

C 192.168.0.0/30 is directly connected, GigabitEthernet0/0/0.20

L 192.168.0.1/32 is directly connected, GigabitEthernet0/0/0.20

D 192.168.0.4/30

[90/3072] via 192.168.0.2, 00:16:21, GigabitEthernet0/0/0.20

D 192.168.0.8/30

[90/3328] via 192.168.0.2, 00:16:17, GigabitEthernet0/0/0.20

R1#sh vrf detail

VRF BGP (VRF Id = 2); default RD 10:1; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0xC

Interfaces:

Gi0/0/0.10 Gi0/0/1.10

Address family ipv4 unicast (Table ID = 0x2):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF EIGRP (VRF Id = 3); default RD <not set>; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0x8

Interfaces:

Gi0/0/0.20 Gi0/0/1.20

Address family ipv4 unicast (Table ID = 0x3):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF Mgmt-intf (VRF Id = 1); default RD <not set>; default VPNID <not set>

New CLI format, supports multiple address-families

Flags: 0x1808

Interfaces:

Gi0

Address family ipv4 unicast (Table ID = 0x1):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast (Table ID = 0x1E000001):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv4 multicast not active

Router 2

R2#show running-config

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf BGP

rd 10:1

ip vrf EIGRP

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO211216BL

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.5 255.255.255.252

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.5 255.255.255.252

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.2 255.255.255.252

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.2 255.255.255.252

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router eigrp 20

address-family ipv4 vrf EIGRP

network 192.168.0.0 0.0.0.3

network 192.168.0.4 0.0.0.3

neighbor 192.168.0.6 GigabitEthernet0/0/0.20

neighbor 192.168.0.1 GigabitEthernet0/0/1.20

autonomous-system 20

eigrp router-id 2.2.2.2

exit-address-family

router bgp 20

bgp router-id 2.2.2.2

bgp log-neighbor-changes

address-family ipv4 vrf BGP

neighbor 192.168.0.1 remote-as 10

neighbor 192.168.0.1 activate

neighbor 192.168.0.6 remote-as 30

neighbor 192.168.0.6 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R2#sh ip route vrf BGP

Routing Table: BGP

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

10.0.0.0/25 is subnetted, 2 subnets

B 10.0.0.0 [20/0] via 192.168.0.1, 00:08:48

B 10.0.0.128 [20/0] via 192.168.0.6, 00:08:05

192.168.0.0/24 is variably subnetted, 4 subnets, 2 masks

C 192.168.0.0/30 is directly connected, GigabitEthernet0/0/1.10

L 192.168.0.2/32 is directly connected, GigabitEthernet0/0/1.10

C 192.168.0.4/30 is directly connected, GigabitEthernet0/0/0.10

L 192.168.0.5/32 is directly connected, GigabitEthernet0/0/0.10

R2#sh ip route vrf EIGRP

Routing Table: EIGRP

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

10.0.0.0/25 is subnetted, 2 subnets

D 10.0.0.0

[90/28416] via 192.168.0.1, 00:15:51, GigabitEthernet0/0/1.20

D 10.0.0.128

[90/28672] via 192.168.0.6, 00:00:42, GigabitEthernet0/0/0.20

192.168.0.0/24 is variably subnetted, 5 subnets, 2 masks

C 192.168.0.0/30 is directly connected, GigabitEthernet0/0/1.20

L 192.168.0.2/32 is directly connected, GigabitEthernet0/0/1.20

C 192.168.0.4/30 is directly connected, GigabitEthernet0/0/0.20

L 192.168.0.5/32 is directly connected, GigabitEthernet0/0/0.20

D 192.168.0.8/30

[90/3072] via 192.168.0.6, 00:14:43, GigabitEthernet0/0/0.20

R2#sh vrf detail

VRF BGP (VRF Id = 2); default RD 10:1; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0xC

Interfaces:

Gi0/0/0.10 Gi0/0/1.10

Address family ipv4 unicast (Table ID = 0x2):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF EIGRP (VRF Id = 3); default RD <not set>; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0x8

Interfaces:

Gi0/0/0.20 Gi0/0/1.20

Address family ipv4 unicast (Table ID = 0x3):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF Mgmt-intf (VRF Id = 1); default RD <not set>; default VPNID <not set>

New CLI format, supports multiple address-families

Flags: 0x1808

Interfaces:

Gi0

Address family ipv4 unicast (Table ID = 0x1):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast (Table ID = 0x1E000001):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv4 multicast not active

Router 3

R3#show running-config

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf BGP

rd 10:1

ip vrf EIGRP

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420G7

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.9 255.255.255.252

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.9 255.255.255.252

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.6 255.255.255.252

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.6 255.255.255.252

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router eigrp 20

address-family ipv4 vrf EIGRP

network 192.168.0.4 0.0.0.3

network 192.168.0.8 0.0.0.3

neighbor 192.168.0.10 GigabitEthernet0/0/0.20

neighbor 192.168.0.5 GigabitEthernet0/0/1.20

autonomous-system 20

eigrp router-id 3.3.3.3

exit-address-family

router bgp 30

bgp router-id 3.3.3.3

bgp log-neighbor-changes

address-family ipv4 vrf BGP

neighbor 192.168.0.5 remote-as 20

neighbor 192.168.0.5 activate

neighbor 192.168.0.10 remote-as 40

neighbor 192.168.0.10 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R3#sh ip route vrf BGP

Routing Table: BGP

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

10.0.0.0/25 is subnetted, 2 subnets

B 10.0.0.0 [20/0] via 192.168.0.5, 00:09:32

B 10.0.0.128 [20/0] via 192.168.0.10, 00:09:32

192.168.0.0/24 is variably subnetted, 4 subnets, 2 masks

C 192.168.0.4/30 is directly connected, GigabitEthernet0/0/1.10

L 192.168.0.6/32 is directly connected, GigabitEthernet0/0/1.10

C 192.168.0.8/30 is directly connected, GigabitEthernet0/0/0.10

L 192.168.0.9/32 is directly connected, GigabitEthernet0/0/0.10

R3#sh ip route vrf EIGRP

Routing Table: EIGRP

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

10.0.0.0/25 is subnetted, 2 subnets

D 10.0.0.0

[90/28672] via 192.168.0.5, 00:17:31, GigabitEthernet0/0/1.20

D 10.0.0.128

[90/28416] via 192.168.0.10, 00:03:29, GigabitEthernet0/0/0.20

192.168.0.0/24 is variably subnetted, 5 subnets, 2 masks

D 192.168.0.0/30

[90/3072] via 192.168.0.5, 00:17:31, GigabitEthernet0/0/1.20

C 192.168.0.4/30 is directly connected, GigabitEthernet0/0/1.20

L 192.168.0.6/32 is directly connected, GigabitEthernet0/0/1.20

C 192.168.0.8/30 is directly connected, GigabitEthernet0/0/0.20

L 192.168.0.9/32 is directly connected, GigabitEthernet0/0/0.20

R3#sh vrf detail

VRF BGP (VRF Id = 2); default RD 10:1; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0xC

Interfaces:

Gi0/0/0.10 Gi0/0/1.10

Address family ipv4 unicast (Table ID = 0x2):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF EIGRP (VRF Id = 3); default RD 20:1; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0xC

Interfaces:

Gi0/0/0.20 Gi0/0/1.20

Address family ipv4 unicast (Table ID = 0x3):

Flags: 0x0

Export VPN route-target communities

RT:20:1

Import VPN route-target communities

RT:20:1

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF Mgmt-intf (VRF Id = 1); default RD <not set>; default VPNID <not set>

New CLI format, supports multiple address-families

Flags: 0x1808

Interfaces:

Gi0

Address family ipv4 unicast (Table ID = 0x1):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast (Table ID = 0x1E000001):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv4 multicast not active

Router 4

R4#show running-config

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R4

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip vrf BGP

rd 10:1

ip vrf EIGRP

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21442B21

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

negotiation auto

interface GigabitEthernet0/0/0.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 10.0.0.254 255.255.255.128

interface GigabitEthernet0/0/0.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 10.0.0.254 255.255.255.128

interface GigabitEthernet0/0/1

no ip address

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 10

ip vrf forwarding BGP

ip address 192.168.0.10 255.255.255.252

interface GigabitEthernet0/0/1.20

encapsulation dot1Q 20

ip vrf forwarding EIGRP

ip address 192.168.0.10 255.255.255.252

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0/2/0

no ip address

shutdown

negotiation auto

interface GigabitEthernet0/2/1

no ip address

shutdown

negotiation auto

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router eigrp 20

address-family ipv4 vrf EIGRP

network 192.168.0.8 0.0.0.3

network 10.0.0.128 mask 0.0.0.127

neighbor 192.168.0.9 GigabitEthernet0/0/1.20

autonomous-system 20

eigrp router-id 4.4.4.4

exit-address-family

router bgp 40

bgp router-id 4.4.4.4

bgp log-neighbor-changes

address-family ipv4 vrf BGP

network 10.0.0.128 mask 255.255.255.128

neighbor 192.168.0.9 remote-as 30

neighbor 192.168.0.9 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R4#sh ip route vrf BGP

Routing Table: BGP

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, 3 subnets, 2 masks

B 10.0.0.0/25 [20/0] via 192.168.0.9, 00:17:24

C 10.0.0.128/25 is directly connected, GigabitEthernet0/0/0.10

L 10.0.0.254/32 is directly connected, GigabitEthernet0/0/0.10

192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.0.8/30 is directly connected, GigabitEthernet0/0/1.10

L 192.168.0.10/32 is directly connected, GigabitEthernet0/0/1.10

R4#sh ip route vrf EIGRP

Routing Table: EIGRP

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, 3 subnets, 2 masks

D 10.0.0.0/25

[90/28928] via 192.168.0.9, 00:18:22, GigabitEthernet0/0/1.20

C 10.0.0.128/25 is directly connected, GigabitEthernet0/0/0.20

L 10.0.0.254/32 is directly connected, GigabitEthernet0/0/0.20

192.168.0.0/24 is variably subnetted, 4 subnets, 2 masks

D 192.168.0.0/30

[90/3328] via 192.168.0.9, 00:18:22, GigabitEthernet0/0/1.20

D 192.168.0.4/30

[90/3072] via 192.168.0.9, 00:18:26, GigabitEthernet0/0/1.20

C 192.168.0.8/30 is directly connected, GigabitEthernet0/0/1.20

L 192.168.0.10/32 is directly connected, GigabitEthernet0/0/1.20

R4#sh vrf detail

VRF BGP (VRF Id = 2); default RD 10:1; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0xC

Interfaces:

Gi0/0/0.10 Gi0/0/1.10

Address family ipv4 unicast (Table ID = 0x2):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF EIGRP (VRF Id = 3); default RD <not set>; default VPNID <not set>

Old CLI format, supports IPv4 only

Flags: 0x8

Interfaces:

Gi0/0/0.20 Gi0/0/1.20

Address family ipv4 unicast (Table ID = 0x3):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast not active

Address family ipv4 multicast not active

VRF Mgmt-intf (VRF Id = 1); default RD <not set>; default VPNID <not set>

New CLI format, supports multiple address-families

Flags: 0x1808

Interfaces:

Gi0

Address family ipv4 unicast (Table ID = 0x1):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv6 unicast (Table ID = 0x1E000001):

Flags: 0x0

No Export VPN route-target communities

No Import VPN route-target communities

No import route-map

No global export route-map

No export route-map

VRF label distribution protocol: not configured

VRF label allocation mode: per-prefix

Address family ipv4 multicast not active

Switch

*(Both Switches have the same configuration)*

SW#show running-config

no service pad

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

hostname S1

boot-start-marker

boot-end-marker

no aaa new-model

system mtu routing 1500

vtp domain CCNP

vtp mode transparent

spanning-tree mode pvst

spanning-tree extend system-id

vlan internal allocation policy ascending

vlan 2

name forleft

vlan 3

name forright

vlan 10

name BGP

vlan 20

name EIGRP

vlan 30

name Expedia

vlan 40

name forty

vlan 100

vlan 996

name CUSTOMER\_NATIVE

interface FastEthernet1/0/1

switchport access vlan 10

switchport mode access

interface FastEthernet1/0/2

switchport access vlan 10

switchport mode access

interface FastEthernet1/0/3

switchport access vlan 10

switchport mode access

interface FastEthernet1/0/4

switchport access vlan 10

switchport mode access

interface FastEthernet1/0/5

switchport access vlan 10

switchport mode access

interface FastEthernet1/0/6

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/7

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/8

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/9

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/10

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/11

switchport access vlan 20

switchport mode access

interface FastEthernet1/0/12

switchport trunk encapsulation dot1q

switchport mode trunk

interface FastEthernet1/0/13

interface FastEthernet1/0/14

interface FastEthernet1/0/15

interface FastEthernet1/0/16

interface FastEthernet1/0/17

interface FastEthernet1/0/18

interface FastEthernet1/0/19

interface FastEthernet1/0/20

interface FastEthernet1/0/21

interface FastEthernet1/0/22

interface FastEthernet1/0/23

interface FastEthernet1/0/24

interface GigabitEthernet1/0/1

interface GigabitEthernet1/0/2

interface GigabitEthernet1/1/1

speed auto 1000

interface GigabitEthernet1/1/2

speed auto 1000

interface Vlan1

no ip address

shutdown

ip http server

ip http secure-server

logging esm config

line con 0

line vty 0 4

login

line vty 5 15

login

end

Conclusion

Like VLANs, VRFs are significant in keeping portions of a network private. I learnt how to set up VRFs for BGP and EIGRP, though I’m sure I could translate the concepts to other routing protocols. At first, I was apprehensive of VRF’s use cases, but I came to appreciate how convenient it could be for larger organizations who may rely on isolated traffic.