Easy Virtual Network—Simplifying Layer 3 Network Virtualization

This paper introduces the new Layer 3 network virtualization solution Easy Virtual Network (EVN). It discusses the need for enterprise network virtualization and compares EVN with the traditional solutions. In-depth architectural information as well as the new provisioning syntax is included to get users fully familiarized with EVN at first look.

Network virtualization is an economical way to provide traffic separation. Multiple virtualized networks can be overlaid on a single physical infrastructure. A corporation may need to provide traffic separation between different user groups. Traffic separation may be based on user role or user group policies. For example, traffic separation may be required between different departments in an organization, third-party vendors may need to share selected network resources, or due to corporation acquisitions and mergers network access may need to be partially restricted. Deploying a separate physical network for each user group increases capital expenditures/operating expenses (CapEx/OpEx) and may not be a viable way to provide traffic separation. Many virtual networks with different security and routing polices can be built over a single physical infrastructure without affecting end users' ability to access needed network resources.

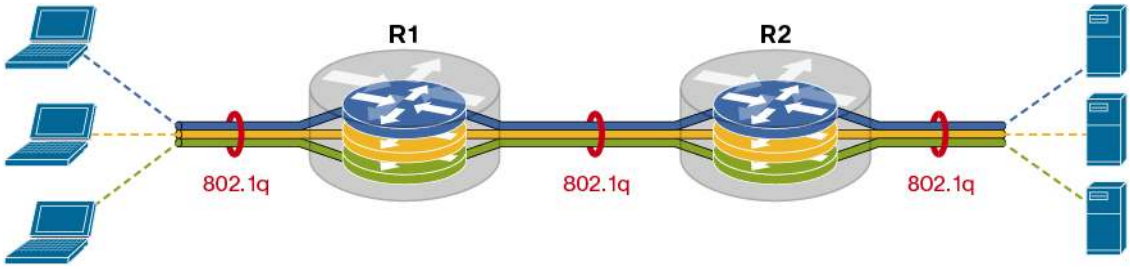
Several well-adopted network virtualization solutions are available in Cisco IOS® Software, for example, Multiprotocol Label Switching (MPLS) VPNs, MPLS VPNs over IP (multipoint generic routing encapsulation [mGRE] and Layer 2 Tunneling Protocol Version 3 [L2TPv3]), Multi-Virtual Route Forwarding (VRF) (also known as VRF-lite), and Policy Based Routing (PBR) or Access Control lLists (ACLs). For some network managers, it may not be desirable to deploy and manage MPLS, Border Gateway Protocol (BGP), networkwide ACLs, or Multi-VRF that require dedicated per virtual network (VN) physical/logical links on a VN path. Networks not requiring more than 32 VNs as well as those that seek an MPLS and BGP-free solution may benefit from an alternate solution.

Easy Virtual Network (EVN) is a simplified LAN virtualization solution that helps enable network managers to provide service separation on a shared network infrastructure. It uses existing technology to increase the effectiveness of VRFs. Existing enterprise network architecture and protocols as well as concepts such as trunk and access interface are preserved in EVN architecture. In addition to reutilizing Multi-VRF features, new components such as VNET trunk, VNET tag, route replication, and management tools are introduced to provide a comprehensive pure IP network segmentation solution.

Multi-VRF offers MP-BGP and label-free network segmentation solution but requires a setup of hop-by-hop path isolation. Separate interfaces or subinterfaces must be provisioned for each virtual network on core-facing interfaces on an end-to-end virtualized path as shown in Figure 1. Network provisioning and management could become repetitive and complex depending on the numbers of virtual networks and numbers of hops traffic needs to cross.

Figure 1 displays three virtual networks: Blue, Yellow, and Green. Notice, there are three separate interfaces dedicated for each between R1 and R2. Blue virtual network traffic is forwarded over the interface/subinterface that is provisioned for the Blue virtual network. This guarantees traffic separation in the forwarding plane. Virtual network devices peer over separate routing instances providing control plane separation. For example, the Blue VRF table holds Blue virtual network routes and the Yellow VRF table holds Yellow virtual network routes.

Figure 1. Multi-VRF Network



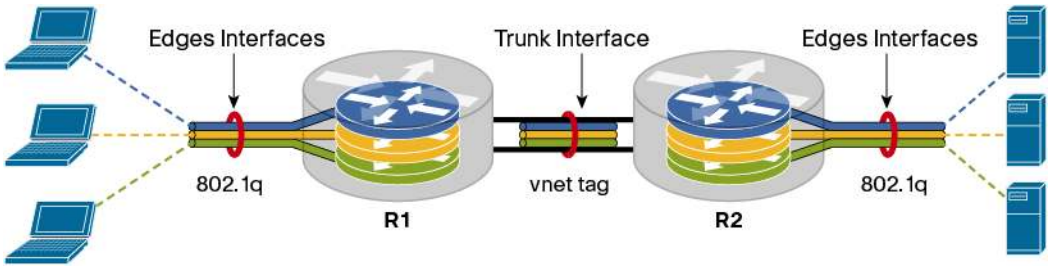
Multi-VRF is manageable for networks with fewer numbers of virtual networks and fewer numbers of hops in a virtual network path. As the numbers of virtual networks grow, new interfaces/subinterfaces will need to be added, and the need for IP addresses and routing will increase. This increases planning and provisioning overhead.

EVN provides the same benefits for guaranteeing traffic separation but introduces an enhanced path isolation technique, simpler provisioning mechanisms, and a lightweight shared services setup to allow communication between VNs.

Traffic Separation in EVN

Path isolation can be achieved by using a unique tag for each VN. This tag is called the VNET tag. Each VN carries over a virtual network the same tag value that was assigned by a network administrator. An EVN device in the virtual path uses the tags to provide traffic separation among different VNs. This removes the dependency on physical/logical interfaces to provide traffic separation. As illustrated in Figure 2, only a single trunk interface is required to connect a pair of EVN devices. A trunk interface provides connectivity between a pair of EVN devices and transports multiple VN traffic, whereas edge interfaces connect to specific VN users. An edge interface is mapped to a specific VN and is the point in the network where the VNET tag is applied to incoming traffic from VN users. Traffic traversing from an EVN device to VN users is untagged. Midpoint EVN devices do not remove, add, or swap tags.

Figure 2. EVN Network



The VNET tag is a global value and thus VNs on each EVN device should be provisioned with the same tag value. The valid tag value range is from 2 to 4094. EVN is supported on any interface that supports 802.1q encapsulation, for example, an Ethernet interface. Instead of adding a new field to carry the VNET tag in a packet, the VLAN ID field in 802.1q is repurposed to carry a VNET tag. The VNET tag uses the same position in the packet as a VLAN ID. On a trunk interface, the packet gets re-encapsulated with a VNET tag. Untagged packets carrying the VLAN ID are not EVN packets and could be transported over the same trunk interfaces.

Static routes, Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF) can be deployed on edge interfaces connecting to VN users or VN sites. EIGRP or OSPF is recommended as the core routing protocol on trunk interfaces.

Each VN runs a separate instance of the routing protocol. Next-hop lookup occurs in the associated VRF table for tagged packets, whereas untagged packets are forwarded using the EVN global routing table. By default, non-VRF interfaces belong to the EVN global table. Notice, the EVN global table is also known as the default routing table. For this reason, although allowed syntactically, it is highly recommended that network managers do not use the VRF name “global” or any variation that uses different cases of letters to define customer VRFs.

Figure 3. Packet Forwarding in an EVN Network

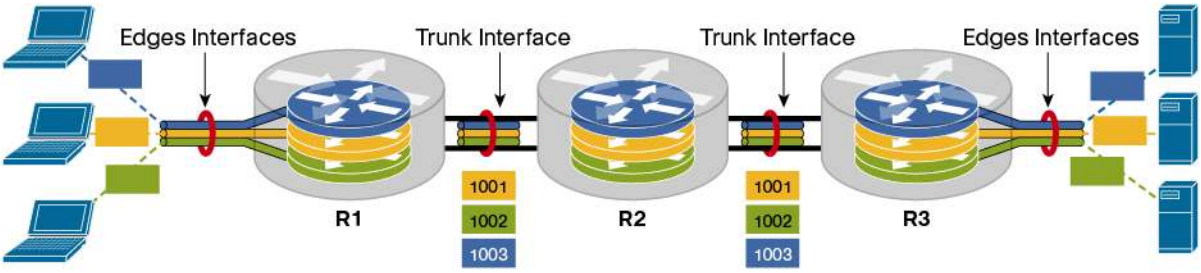


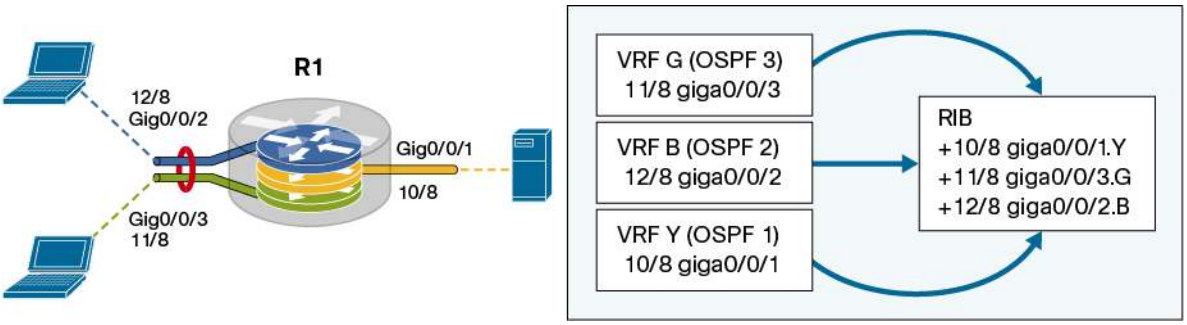
Figure 3 illustrates packet forwarding in an EVN network:

1. Untagged packets arrive on the ingress edge interface of R3. The ingress edge interface is associated with a particular VRF, for example, VRF Blue. EVN Blue is preconfigured with a tag value 1003.
   - R3 does route lookup in VRF Blue, and the next hop is R2 through a trunk interface.
   - R3 encapsulates the packet with VNET Blue tag 1003 and forwards it over the trunk interface to R2.
2. R2 receives the packet on a trunk interface. It uses tag 1003 to identify that the packet belongs to VRF Blue.
   - R2 does route lookup in VRF Blue, and the next hop is R1 through the trunk interface.
   - R2 encapsulates the packet with VNET Blue tag 1003 and forwards it over the trunk interface to R1.
3. R1 receives the packet on the trunk interface. It uses tag 1003 to identify that the packet belongs to VRF Blue.
   - R1 does route lookup in VRF Blue, and the next hop is through the edge interface.
   - R1 sends an untagged packet over the edge interface.
4. User receives the untagged packet.

Supporting Shared Services in an EVN network

Common sets of services such as Internet connectivity, email, video, Dynamic Host Configuration Protocol (DHCP), or Domain Name System (DNS) may be required by multiple EVN users. Shared services help enable clients of one VN to access services located in other VNs. This is also known as extranet or overlapping VNs. To allow communication between clients and servers of different VNs, network Layer Layer Reachability Information (NLRI) must be made available among VNs. Multi-VRF achieves route exchange between VNs by redistributing VN routes indirectly through BGP using the route-target import/export feature. The route replication feature in EVN removes the dependency on BGP and any additional BGP attributes such as route target and route distinguisher. Route replication allows each EVN to have direct access to the Routing Information Base (RIB), removing the need to duplicate routing tables or routes, and saving CPU and memory. Route redistribution is not required between VNs for the users and services connected to a local router. Where possible, to eliminate the need of redistributing routes, the recommendation is to implement route replication on a router that is directly connected to the server subnet as illustrated in Figure 4.

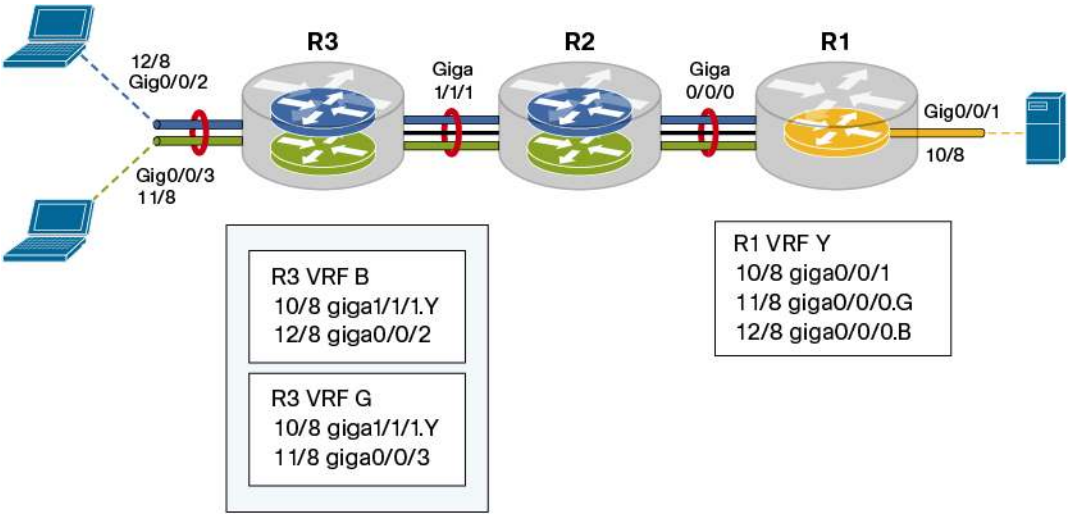
Figure 4. EVN Routing Tables After Route Replication



Notice, server prefix 10/8 is replicated into each host EVN. Replicated routes marked with the plus sign (+) will have the same administrative distance as the original route. Route replication only replicates prefixes and not the next hop. A route can be replicated only once.

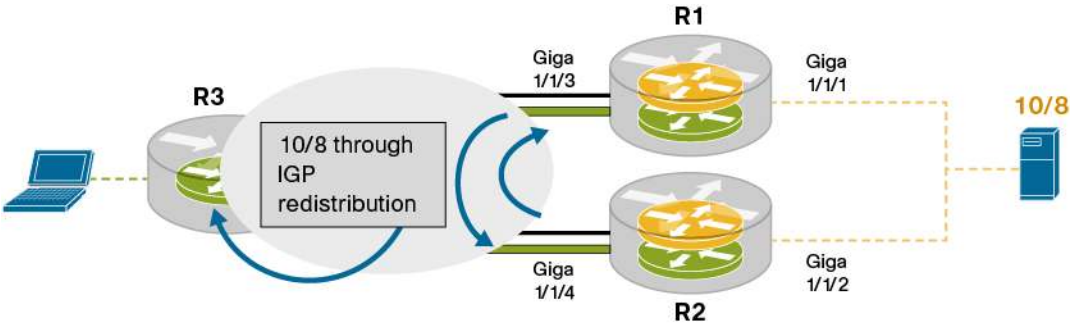
Replicated routes are not redistributed automatically. Route redistribution into Interior Gateway Protocol (IGP) or Exterior Gateway Protocol (EGP) is required after route replication to propagate routes across each EVN so that remote users can access shared services. Figure 5 demonstrates a scenario where Yellow VN routes are replicated and then redistributed.

Figure 5. EVN Routing Tables After Route Replication and Redistribution



Building a fault-tolerant network with redundant components is critical to help minimize packet loss and assure continuous service availability and seamless disaster recovery. Thus it's typical for a network to have redundancy at fusion points connecting to shared services. This may introduce a situation where replicated routes are redistributed at multiple points in the network generating route oscillation. Route oscillation causes a redistribution point router to continuously switch between two or more candidate routes. To prevent route oscillation, a replicated route is assigned an administrative distance value of zero and is always preferred over an IGP route. For the topology shown in Figure 6, the Yellow VN service prefix 10/8 is replicated and redistributed so that Green VN users can access the service. The R1 RIB will have 10/8 as a replicated route as well as a redistributed route from R2 and vice versa. The best route selected on R1 to reach the Yellow VN will be through giga1/1/1 and on R2 through giga1/1/2. However, the IGP metric should be assigned accordingly on R1 and R2 so that traffic is forwarded over a preferred path from Green VN users to the services segment.

Figure 6. Multiple Replication and Redistribution Points



When there are multiple sources for the same route and the shared services segment is multiple hops away from the router doing route replication, the routing table uses the following preference rules to choose the route for installation:

1. Prefers a route with a better IGP administrative distance
2. Prefers a route with a better default administrative distance
3. Prefers a nonreplicated route over a replicated route
4. Prefers a replicated route with a better replicate distance
5. Prefers a replicated route with a better lexical value of the source VRF name

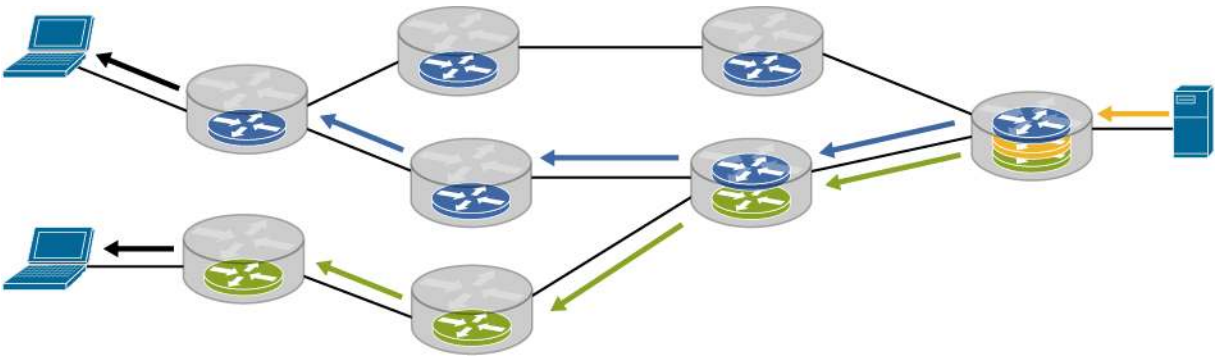
Shared IPv4 Multicast VN Service

Multicast is an integral part of many networks. EVN for multicast VNs provides the capability of supporting multicast VRFs. In a shared multicast service environment, receivers of different multicast VNs need to receive traffic from the same multicast source. To facilitate this, (S,G) needs to be created such that its Reverse Path Forwarding interface and members in the outgoing interface list belong to a different EVN. EVN offers two different approaches to support Source Specific Multicast mode for extranet service:

1. Multicast traffic in all EVNs
2. Multicast traffic in multicast EVNs

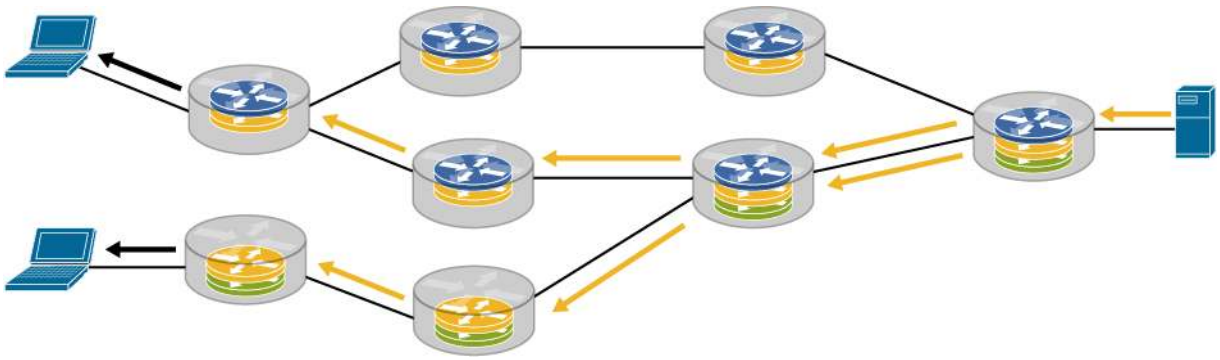
As shown in Figure 7 (for the first approach listed above), the server prefix is replicated into each mVRF. Joins from the receivers on different mVRFs are forwarded toward the first-hop router closest to the server. At the first-hop router, individual (S,G) for each mVRF are merged to create a combined (S,G) state, which is used to replicate multicast traffic onto multiple EVNs.

Figure 7. Multicast Data Traffic Flow When the First-Hop EVN Router Replicates Traffics



Alternately, a dedicated source multicast EVN containing the server prefix could be replicated into other EVNs on the first-hop router connected to the server. On the last-hop routers connecting the receivers, the joins are sent in the users' multicast EVN. As a result, a combined shared tree is built in the multicast EVN for all receivers. As shown in Figure 8, on the first-hop router closest to the server, the (S,G) state will put multicast traffic on the multicast EVN. Traffic stays in the multicast EVN until it reaches the last-hop router, where the (S,G) will send the traffic to the receiver over the edge interface.

Figure 8. Multicast Data Traffic Flow When Multicast Virtual Network Is Used



Migrating from Multi-VRF to EVN

Several enterprise networks have deployed Multi-VRF. Some of these networks may be expanding or adding more VNs that can take advantage of EVN. The long-term recommended strategy would be to deploy a single segmentation solution, but during the migration phase it may be necessary to support both Multi-VRF and EVN. The EVN command-line interface (CLI) uses the existing VRF CLI commands and, as a result, allows any existing VRF CLI to work on EVN without changes. Thus EVN and Multi-VRF deployments can coexist in the same network and on the same router.

IPv6 in EVN

IPv4 was the main Layer 3 protocol in most of the enterprise networks. IPv6 is now mandated by the federal government for compliance across all organizations. Furthermore, exhaustion of the IPv4 address pool is driving several networks to start deploying IPv6. Some IPv6 networks may also require IPv6 VN service. As a result, IPv6/IPv4 global and VN traffic needs to coexist. In some cases this creates a need to support dual stack protocols on the same access interface connecting to VN users. Non-IPv6 traffic traveling on the same network path as IPv4 VNs can be transported through the EVN global table on trunk interfaces. Multi-VRF is available to support IPv6 VRFs. Networks that have dual stack VNs on access links, Multi-VRF for IPv6 VNs, and EVN for IPv4 VNs can be deployed concurrently.

Simplified Provisioning in EVN

Basic VN Configuration

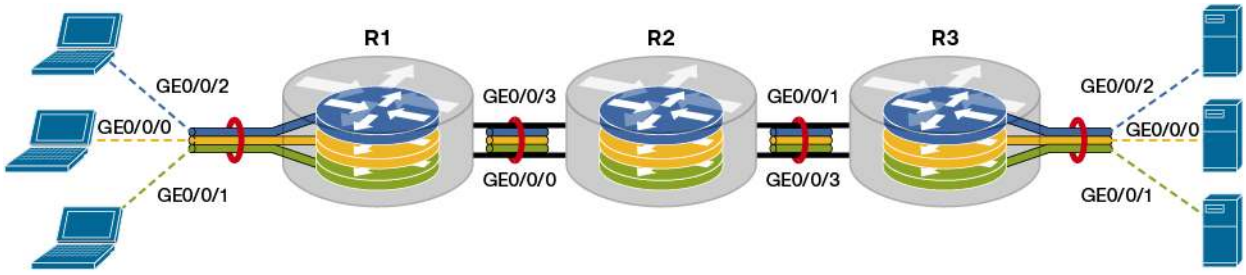
Both EVN and VRF (for Multi-VRF or other MPLS Layer 3 VNs) could coexist on the same router. The EVN CLI uses the existing VRF CLI commands and, as a result, allows any existing VRF CLI to work on EVN without changes. The following CLI command defines the Blue VN:

```
!
vrf definition Blue
! Configures the Blue VN. Notice: the VN name is case sensitive!
vnet tag 1003
! Assigns an EVN tag value of 1003 to the Blue VN. Notice this is different from
Multi-VRF configuration. Multi-VRF doesn't support the VN tag!
address-family ipv4
! Declares carrying the IPv4 prefixes!
interface giga0/0/2
 vrf forwarding Blue
 ip address 10.1.3.1 255.255.255.0
! Sets up an edge interface connecting to Blue VN users!
interface giga0/0/3
vnet trunk
ip address 10.1.10.1 255.255.255.0
! Sets up the core-facing trunk interface. The trunk interface connects to another
EVN router. Notice the difference compared to Multi-VRF configuration. Multi-VRF
requires the vrf forwarding command on every core-facing interface!
```

Notice a single trunk interface transporting multiple EVN traffic doesn't require the *vrf forwarding* command.

EVN configuration for the topology is displayed in Figure 9.

Figure 9. EVN Configuration for Basic VN



Hostname R1	Hostname R2	Hostname R3
!	!	!
vrf definition Yellow	vrf definition Yellow	vrf definition Yellow
vnet tag 1001	vnet tag 1001	vnet tag 1001
address-family ipv4	address-family ipv4	address-family ipv4
!	!	!
vrf definition Green	vrf definition Green	vrf definition Green
vnet tag 1002	vnet tag 1002	vnet tag 1002
address-family ipv4	address-family ipv4	address-family ipv4
!	!	!
vrf definition Blue	vrf definition Blue	vrf definition Blue
vnet tag 1003	vnet tag 1003	vnet tag 1003
address-family ipv4	address-family ipv4	address-family ipv4
!	!	!
int giga0/0/0	int giga0/0/0	int giga0/0/0
vrf forwarding Yellow	vnet trunk	vrf forwarding Yellow
ip address 10.1.1.1 255.255.255.0	ip address 10.1.10.2 255.255.255.0	ip address 10.1.4.1 255.255.255.0
! int giga0/0/1	!	! int giga0/0/1
vrf forwarding Green	int giga0/0/1	vrf forwarding Green
ip address 10.1.2.1 255.255.255.0	vnet trunk	ip address 10.1.5.1 255.255.255.0
!	ip address 10.1.11.2 255.255.255.0	!
int giga0/0/2	!	int giga0/0/2
vrf forwarding Blue		vrf forwarding Blue
ip address 10.1.3.1 255.255.255.0		ip address 10.1.6.1 255.255.255.0
!		!
int giga0/0/3		int giga0/0/3
vnet trunk		vnet trunk
ip address 10.1.10.1 255.255.255.0		ip address 10.1.11.1 255.255.255.0
!		!

Comparing Multi-VRF and EVN CLI

As shown in the following configuration Multi-VRF doesn't have a trunk interface; each subinterface for each VRF has to be configured manually. EVN automatically generates subinterfaces for each EVN that do not expand in the configuration to keep the configuration concise.

Multi-VRF Core Interface Configuration	EVN Core Interface Configuration
! interface TenGigabitEthernet1/1 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 TenGigabitEthernet1/1.101** 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 TenGigabitEthernet1/1.102** 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 !	! vrf definition Red **vnet tag 101** vrf definition Green **vnet tag 102** ! interface TenGigabitEthernet1/1 **vnet trunk** ip address 10.122.5.32 255.255.255.254 ip pim query-interval 333 msec ip pim sparse-mode logging event link-status !

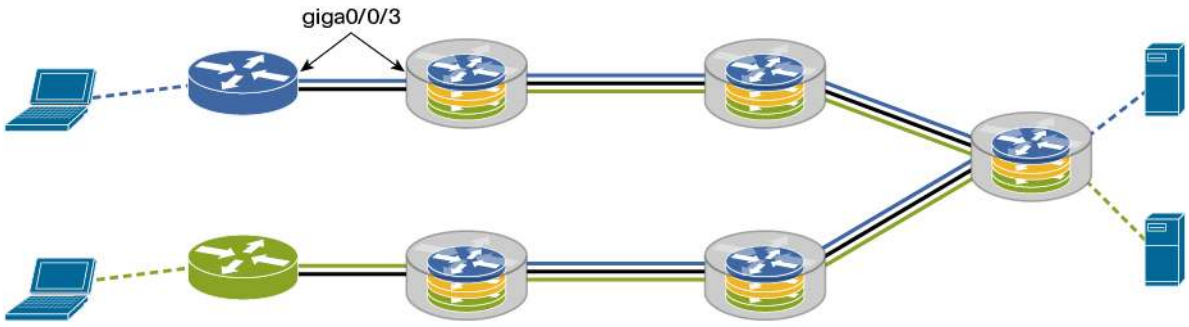
Restricting VN Traffic over a Trunk Interface

A trunk interface transports all VN traffic. The network may require a trunk interface to restrict the VNs it transports. The reason for restriction may be that only a single VN site is connected to an EVN device. Or there may be a need to define an explicit path if multiple paths between a VN endpoint exist.

The following example allows Blue VN traffic to be transported over a trunk interface as shown in Figure 10:

```
!
vrf list AllowB
 member Blue
!
int giga0/0/3
   vnet trunk list AllowB
!
```

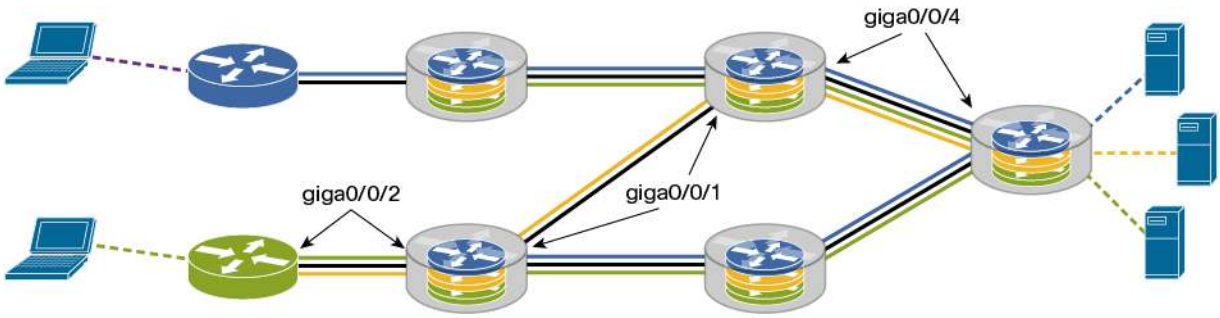
Figure 10. Packet Forwarding in an EVN Network



The following example allows Yellow VN traffic to be transported over the middle path in the network. Only trunk interface giga0/0/1 and giga0/0/2 and giga0/0/4 should allow the Yellow VN, as shown in Figure 11.

```
!
vrf list AllowY
 member Yellow
!
vrf list AllowYG
 member Yellow
 member Green
!
vrf list AllowYGB
 member Yellow
 member Green
 member Blue
!
int giga0/0/1
   vnet trunk list AllowY
!
int giga0/0/2
   vnet trunk list AllowYG
!
int giga0/0/4
   vnet trunk list AllowYGB
!
```

Figure 11. Packet Forwarding in an EVN Network over an Explicit Path



Customizing Trunk Interface Attributes

A network may require the customization of interface attributes such as bandwidth, next hop, delay, hold-time, and so on relevant to global routing table. The following CLI command allows users to change the default OSPF cost:

```
int g1/1
  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
```

Notice, global EVN is relevant only on a trunk interface; thus provisioning is done under a trunk interface. Provisioning of global EVN is not required as it's automatically created by an EVN device.

Provisioning Shared Services

Route replication in EVN allows sharing of common services among different VNs. The recommendation is to enable route replication on the router directly connected to a shared services segment. The following CLI command enables route replication where the EVN Blue and Green VNs contain clients whereas the shared service EVN Yellow contains the servers.

```
!
vrf definition Yellow
  vnet tag 1001
  address-family ipv4
    route-replicate from vrf Green all route-map server-prefix-map
   route-replicate from vrf Blue all route-map server-prefix-map
!
vrf definition Green
  vnet tag 1002
  address-family ipv4
    route-replicate from vrf Yellow all route-map server-prefix-map
!
vrf definition Blue
  vnet tag 1003
  address-family ipv4
    route-replicate from vrf Yellow all route-map server-prefix-map
!
```

Setting Up Shared Services for Global Users

A service can also be shared with global users by replicating it into the global table using the following CLI command:

```
!
  Global-address-family ipv4unicast
    route-replicate from vrf Yellow all route-map server-prefix-map
 !
```

EVN Network Verification and Management

Network managers need to access VN information during initial service provisioning to help assure that service is set up and working as expected. Alternatively, in the live network, to promptly respond to network failures and minimize service downtime, network managers need access to VN information periodically. Simple yet effective tools are essential to help verify service availability, troubleshoot problems in the signaling or forwarding planes, and collect statistics on VN interfaces. Routing context, traceroute, debug condition, and cisco-vrf-mib facilitate simpler network management and troubleshooting.

Routing context supports viewing or resetting of relevant components under each EVN, reducing the repetition of commands required to specify an EVN of interest. Debug, ping, traceroute, show and clear functionality for vNET Manager, IP, Address Resolution Protocol (ARP), OSPF, EIGRP, Protocol Independent Multicast (PIM), Internet

Group Management Protocol (IGMP), and Simple Network Management Protocol (SNMP) are some examples. The following command allows access from routing context for the Yellow VRF:

```
R1# routing-context vrf Yellow
R1%Yellow# show ip ospf neighbor (!Displays OSPF neighbour for Yellow EVN)
Neighbor ID     Pri   State           Dead Time   Address         Interface
125.0.12.16       1   FULL/BDR        00:00:30    125.1.16.16     Ethernet3/0.12

R1%Yellow# show ip route   (!Displays EVN Yellow routes)
...
O E2     125.0.10.0/24 [110/1] via 125.1.16.16, 1w3d, Ethernet3/0.12
O E2     125.0.10.11/32 [110/11] via 125.1.16.16, 1w3d, Ethernet3/0.12
...
```

Debug messages are used to examine states and activities between peers or local router service components. A router can have up to 32 EVNs. It is important to filter out messages pertaining to all other EVNs than the one a network manager is interested in examining. The debug condition feature allows users to set an EVN debug context that limits generating debug messages. Debug condition offers a flexible model and allows the setting of preferred debugs at the global level or routing context level. The following example generates debug output for the VRF Red OSPF instance only:

```
R1# debug condition vrf Red
R1# debug ip ospf hello
R1# debug ip ospf spf
```

The same can be specified under vrf red routing context:

```
R1%Red#deb condition vrf Red
Condition 1 set
CEF filter table debugging is on
R1%Red#
*Oct 25 23:10:02.927: vrfmgr(0) Debug: Condition 1, vrf Red triggered, count 1
```

Explicitly specified debug takes precedence over a specific debug condition setting. As demonstrated in the following example, the second command would take effect for VRF Blue even though the debug condition for VRF Red was specified:

```
R1# debug condition vrf Red
R1# debug ip eigrp vrf Blue neighbor
```

Debugs for a nonmatching VRF can be set under any other VRF's routing context. The following example enables debugs for the Blue VRF from the Red VRF routing context:

```
R1%Red# deb condition vrf Blue
Condition 2 set
CEF filter table debugging is on
R1%Red#
*Oct 25 23:12:09.990: vrfmgr(1) Debug: Condition 2, vrf Blue triggered, count 1
```

Exactly which VRF's debug conditions were set can be verified using the **show debug condition** command. The following log shows debug condition for VRFs Red and Blue were set:

```
R1%Red#sh deb condition

Condition 1: vrf Red (1 flags triggered)
        Flags: vrfmgr(0)
Condition 2: vrf Blue (1 flags triggered)
        Flags: vrfmgr(1)
```

Debug condition can be disabled for a selected VRF:

```
R1%Red#no debug condition vrf Red
Condition 1 has been removed
```

Alternatively, debug condition can be disabled for all VRFs with a single command:

```
R1%Red#no debug condition all
Removing all conditions may cause a flood of debugging
messages to result, unless specific debugging flags
are first removed.

Proceed with the removal of all conditions? [yes/no]:y
2 conditions have been removed
```

The ping utility is useful in verifying the accessibility of VN prefixes. The ping utility has been modified to be supported for each EVN. For example, a prefix's reachability can be checked under an EVN using the routing context. The following example uses the ping utility under a routing context:

```
R1# routing-context vrf Yellow
R1%Yellow# ping 125.0.10.11
Sending 5, 100-byte ICMP Echos to 125.0.10.11, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The traceroute utility provides information on a hop-by-hop path traversed by traffic to reach the destination. Since EVN traffic will be transported over shared core links and IP addresses, the traceroute utility has been enhanced to include EVN tags relevant for each VRF.

```
R1%Red# traceroute 10.1.3.1
1   10.1.1.2   [Incoming: Red, Outgoing: Red:1001]
2   10.2.1.2   [Incoming: Red:1001, Outgoing: Red:1001]
```

Each trunk interface carries traffic for multiple EVNs. The trunk interface is implemented as a group of hidden subinterfaces, one subinterface for each EVN. Each subinterface is automatically created using the format <*interface*>.<*EVN name*>. Notice, EVN subinterface configuration on a trunk interface doesn't expand on the router, which makes it impossible to identify a subinterface number created for an EVN through the *show running* or *show configuration* command. Each EVN interface's information can be retrieved using *show interface <#.EVN name>* or

show interface<#.vNET-tag>. Alternately, the cisco-vrf-mib MIB can be used by SNMP to collect VN information. The following example shows logs on a trunk interface Ethernet 1/0 for EVN Red using VNET tag 13:

```
R1#show interface   ethernet 1/0.13
Ethernet1/0.13 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.0b01 (bia aabb.cc00.0b01)
  Description: Subinterface for VNET Red
  Internet address is 125.1.1.11/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  13.
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive set (10 sec)
  Last clearing of "show interface" counters never
```

Conclusion

Network virtualization solution EVN:

- Provides a pure IP alternative to MPLS in enterprise networks for up to 32 VNs
- Uses existing enterprise design/architecture/protocols
- Uses existing technology to increase the effectiveness of VRFs
- Provides either an IGP (OSPF, EIGRP) only or IGP/EGP-based alternative Reintroduces familiar concepts for access and trunks to Layer 3
- Can be deployed with traditional MPLS VPNs or MPLS VPNs over mGRE
- Can coexist with Multi-VRF deployments
- Supports non-IP and IPv6 traffic through the EVN global table
- Supports PIM and IGMP with SM and SSM modes for mVPN
- Supports shared services using route replication
- Includes enhanced troubleshooting and usability tools:
  - routing context, traceroute, debug condition, cisco-vrf-mib, and simplified VRF-aware SNMP configuration

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-638769-00 01/11