Configuring Segmentation (VPNs)

In the Viptela overlay network, VPNs divide the network into different segments. By default, two VPNs are present in the configurations of all Viptela devices, and these VPNs serve specific purposes:

- VPN 0 is the transport VPN. It carries control traffic over secure DTLS or TLS connections between vSmart controllers and vEdge routers, and between vSmart controllers and vBond orchestrators. Initially, VPN 0 contains all a device's interfaces except for the management interface, and all the interfaces are disabled. For the control plane to establish itself so that the overlay network can function, you must configure WAN transport interfaces in VPN 0.

- VPN 512 is the management VPN. It carries out-of-band network management traffic among the Viptela devices in the overlay network. By default, VPN 512 is configured and enabled. You can modify this configuration if desired.

To segment user networks and user data traffic locally at each site and to interconnect user sites across the overlay network, you create additional VPNs on vEdge routers. (These VPNs are identified by a number that is not 0 or 512.) To enable the flow of data traffic, you associate interfaces with each VPN, assigning an IP address to each interface. These interfaces connect to local-site networks, not to WAN transport clouds. For each of these VPN, you can set other interface-specific properties, and you can configure features specific for the user segment, such as BGP and OSPF routing, VRRP, QoS, traffic shaping, and policing.

This article provides basic configuration procedures for the three types of VPNs. For more detailed information about configuring other VPN parameters, such as interface properties and routing protocols, refer to the Additional Information at the end of this article.

Configure the Transport VPN (VPN 0)

For the control plane to establish itself so that the overlay network can function, you must configure interfaces in VPN 0 to carry the control traffic necessary to establish and maintain the overlay network.

On vEdge routers, the interfaces in VPN 0 connect to some type of transport network or cloud, such as the Internet, MPLS, or Metro Ethernet. For each interface in VPN 0, you must set an IP address, and you create a tunnel connection that sets the color and encapsulation for the WAN transport connection. (The encapsulation is used for the transmission of data traffic.) These three parameters—IP address, color, and encapsulation—define a TLOC (transport location) on the vEdge router. The OMP session running on each tunnel sends the TLOC to the vSmart controllers so that they can learn the overlay network topology. For VPN 0, you can also set other interface-specific and VPN-specific properties in VPN 0.

Because vSmart controllers are responsible for determining the best routes through the overlay network (based on the TLOCs it learns and based on centralized policies), they handle only control plane traffic, in VPN 0. A vSmart controller can have only one interface in VPN 0, for which you set an IP address and you create a tunnel connection. This tunnel connection acts a control plane tunnel termination point.
Configure the Transport VPN on a vEdge Router

On a vEdge router, the interfaces in VPN 0 connect to a WAN transport network. You must configure at least one tunnel interface on a vEdge router so that it can join the control plane and be part of the overlay network. If it is not configured, that router cannot participate in the overlay network.

For a tunnel connection on a vEdge router, you must configure the three components of a TLOC—the interface’s IP address and the tunnel’s color and encapsulation. An OMP session runs over each tunnel connection, and it is OMP that distributes the device TLOCs to vSmart controllers. The controllers use the TLOCs to determine the overlay network topology and to determine the best routing paths across the overlay network. A vEdge router can have up to four TLOCs, so you can configure more than one tunnel connection.

In the transport VPN (VPN 0), vEdge routers support dual stack. To enable dual stack, configure an IPv4 address and an IPv6 address on the tunnel interface. The vEdge router learns from the vSmart controller whether a destination supports IPv4 or IPv6 addresses. When forwarding traffic, the router chooses either the IPv4 or the IPv6 TLOC based on the destination address.

To configure VPN 0 on a vEdge router:

1. Configure the WAN transport interface:
   ```
   vEdge(config)# vpn 0 interface interface-name
   vEdge(config-interface)#
   ```
   In the most common cases, `interface-name` is the name of a physical Gigabit Ethernet interface (geport/slot). The interface name can also be `grenumber`, `ipsecnumber`, `loopbackstring`, `natpoolnumber`, or `pppnumber`.

2. Configure a static IPv4 address for the interface:
   ```
   vEdge(config-interface)# ip address prefix/length
   vEdge(config-interface)#
   ```
   Or you can enable DHCP on the interface so that the interface learn its IP address dynamically:
   ```
   vEdge(config-interface)# ip dhcp-client [dhcp-distance number]
   vEdge(config-interface)#
   ```
   When an interface learns its IPv4 address from a DHCP server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the `dhcp-distance` option, specifying a distance from 1 through 255.

3. To enable dual stack, configure a static IPv6 address for the interface:
   ```
   vEdge(config-interface)# ipv6 address prefix/length
   vEdge(config-interface)#
   ```
   Or you can enable DHCPv6 on the interface so that the interface learn its IPv6 address dynamically:
   ```
   vEdge(config-interface)# ipv6 dhcp-client [dhcp-distance number] [dhcp-rapid-commit]
   vEdge(config-interface)#
   ```
   When an interface learns its IPv6 address from a DHCPv6 server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the `dhcp-distance` option, specifying a distance from 1 through 255. To speed up the assignment of IPv6 addresses, include the `dhcp-rapid-commit` option.

4. Enable the interface:
   ```
   vEdge(config-interface)# no shutdown
   ```
5. Configure the WAN transport tunnel connection:
   vEdge(config-interface)# tunnel-interface
   vEdge(config-tunnel-interface)#

6. Configure a color for the tunnel connection as an identifier for the tunnel:
   vEdge(config-tunnel-interface)# color color
   vEdge(config-tunnel-interface)#
   color can be 3g, biz-internet, blue, bronze, custom1, custom2, custom3, default, gold, green, lte, metro-ethernet, mpls, private1 through private6, public-internet, red, and silver. The default color is default. The colors metro-ethernet, mpls, and private1 through private6 are referred to as private colors, because they use private addresses to connect to the remote side vEdge router in a private network. You can use these colors in a public network provided that there is no NAT device between the local and remote vEdge routers.

7. Configure the encapsulation to use on tunnel connection:
   vEdge(config-tunnel-interface)# encapsulation (gre | ipsec)
   vEdge(config-tunnel-interface)#
   To configure both IPsec and GRE encapsulation, include two encapsulation commands. Note that if you do this, you are creating two TLOCs that have the same IP addresses and colors, but that have different encapsulation.

8. Configure any other properties specific to the tunnel interface, the interface, or VPN 0.

9. If you have a multi-TLOC environment, configure additional tunnel interfaces.

10. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from VPN 0:
    vEdge(config-vpn-0)# dns ip-address (primary | secondary)
    The address can be either an IPv4 or IPv6 address. By default, the IP address is for the primary DNS server.

11. If desired, configure IPv4 and IPv6 static routes in VPN 0:
    vEdge(config-vpn-0)# ip route prefix/length next-hop [administrative-distance]
    vEdge(config-vpn-0)# ipv6 route prefix/length next-hop [administrative-distance]

12. Activate the configuration:
    vEdge(config)# commit

To display interface information, use the show interface command for IPv4 interfaces and show ipv6 interfaces for IPv6 interfaces. To display information about DHCP and DHCPv6 servers, use the show dhcp interface and show ipv6 dhcp interface commands.

When you are troubleshooting routing and forwarding problems on a vEdge router, you can configure the router to perform route consistency checks, to determine whether the routes in the router's route and forwarding tables are consistent:

vEdge(config-system)# route-consistency-check

This command checks only IPv4 routes. Route consistency checking requires a large amount of device CPU, so it is recommended that you enable it only when you trouble shooting an issue and that you disable it at other times.

Here is an example of a VPN 0 configuration, where interface ge0/0 is the WAN transport interface. This example shows that dual stack is enabled on the router, because the tunnel interface has both an IPv4 and an IPv6 address.
Notice that the remaining seven device interfaces are part of VPN 0, because we have not yet configured any other VPNs. Also notice that the management interface is not present in VPN 0.

```
vpn 0
  interface ge0/0
  ip address 10.0.0.8/24
  ipv6 address fd00:1234::/16
  tunnel-interface
  color biz-internet
  encapsulation ipsec
  allow-service dhcp
  allow-service dns
  allow-service icmp
  no allow-service sshd
  no allow-service ntp
  no allow-service stun
  !
  no shutdown
  !
  interface ge0/1
  shutdown
  !
  interface ge0/2
  shutdown
  !
  interface ge0/3
  shutdown
  !
  interface ge0/4
  shutdown
  !
  interface ge0/5
  shutdown
  !
  interface ge0/6
  shutdown
  !
  interface ge0/7
  shutdown
  !
```

An interface can participate only in one VPN. So in an initial configuration, when VPN 0 is the only VPN that is configured, all the device’s interfaces are present, by default, in VPN 0 (as shown in the output above). Then, when you create other VPNs to carry data traffic and configure interfaces in those VPNs, the interfaces used in the other VPNs are automatically removed from VPN 0. Here is an example in which `interface ge0/3` is used for VPN 1, so it has been automatically removed from the configuration of VPN 0:

```
vpn 0
  interface ge0/0
  ip address 10.0.0.8/24
  tunnel-interface
  color biz-internet
  encapsulation ipsec
  allow-service dhcp
  allow-service dns
  allow-service icmp
  no allow-service sshd
  no allow-service ntp
  no allow-service stun
  !
  no shutdown
  !
  interface ge0/1
  shutdown
  !
  interface ge0/2
  shutdown
  !
  interface ge0/4
  shutdown
  !
  interface ge0/5
  shutdown
  !
  interface ge0/6
  shutdown
  !
```

https://sdwan-docs.cisco.com/Product_Documentation/Software_Features/Release_18.1/04Segmentation/02Configuring_Segm
When you configure subinterfaces in a VPN that carries data traffic (that is, not VPN 0 and not VPN 512), the main interface must be configured with the `no shutdown` command so that it is enabled, and the main interface remains in VPN 0 once you configure the subinterface. For example, if in the VPN 1 configuration, you were to configure OSPF on VLAN 1, you can see that `interface ge0/3` remains present in VPN 0, while the subinterface `interface ge0/3.1` is used in VPN1:
Configure the Transport VPN on a vSmart Controller

Because vSmart controllers are responsible for determining the best routes through the overlay network (based on the TLOCs it learns and based on centralized policies), they handle only control plane traffic, in VPN 0. A vSmart controller can have only one interface in VPN 0, for which you set an IP address and you create a tunnel connection. This tunnel connection acts a control plane tunnel termination point.

In the transport VPN (VPN 0), vEdge routers support dual stack. To enable dual stack, configure an IPv4 address and an IPv6 address on the tunnel interface. The vEdge router learns from the vSmart controller whether a destination supports IPv4 or IPv6 addresses. When forwarding traffic, the router chooses either the IPv4 or the IPv6 TLOC based on the destination address.

To configure VPN 0 on a vSmart controller:

1. Configure the WAN transport interface:
   ```
   vSmart(config)# vpn 0 interface interface-name
   vSmart(config-interface)#
   interface-name is the name of a virtual Ethernet interface (ethnumber).
   ```

2. Configure a static IPv4 address for the interface:
   ```
   vEdge(config-interface)# ip address prefix/length
   vEdge(config-interface)#
   Or you can enable DHCP on the interface so that the interface learn its IP address dynamically:
   vEdge(config-interface)# ip dhcp-client [dhcp-distance number]
   vEdge(config-interface)#
   When an interface learns its IPv4 address from a DHCP server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the dhcp-distance option, specifying a distance from 1 through 255.
   ```

3. To enable dual stack, configure a static IPv6 address for the interface:
   ```
   vEdge(config-interface)# ipv6 address prefix/length
   vEdge(config-interface)#
   Or you can enable DHCPv6 on the interface so that the interface learn its IP address dynamically:
   vEdge(config-interface)# ipv6 dhcp-client [dhcp-distance number] [dhcp-rapid-commit]
   vEdge(config-interface)#
   When an interface learns its IPv6 address from a DHCPv6 server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the dhcp-distance option, specifying a distance from 1 through 255. To speed up the assignment of IPv6 addresses, include the dhcp-rapid-commit option.
   ```

4. Enable the interface:
   ```
   vSmart(config-interface)# no shutdown
   ```

5. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from VPN 0:
   ```
   vEdge(config-vpn-0)# dns ip-address (primary | secondary)
   The address can be either an IPv4 or IPv6 address. By default, the IP address is for the primary DNS server.
   ```
6. If desired, configure IPv4 and IPv6 static routes in VPN 0:
   ```
   vEdge(config-vpn-0)# ip route prefix/length next-hop [administrative-distance]
   vEdge(config-vpn-0)# ipv6 route prefix/length next-hop [administrative-distance]
   ```

7. Configure any other properties specific to the tunnel interface, the interface, or VPN 0.

8. Activate the configuration:
   ```
   vSmart(config)# commit
   ```

To display interface information, use the `show interface` command for IPv4 interfaces and `show ipv6 interfaces` for IPv6 interfaces. To display information about DHCP and DHCPv6 servers, use the `show dhcp interface` and `show ipv6 dhcp interface` commands.

Here is an example of a VPN 0 configuration on a vSmart controller:

```
vSmart# show running-config vpn 0
vpn 0
dns 1.2.3.4 primary
interface eth0
  ip dhcp-client
  no shutdown
!
interface eth1
  ip address 10.0.5.19/24
tunnel-interface
  allow-ssh
  allow-icmp
  no shutdown
!
ip route 0.0.0.0/0 10.0.5.13
!
```

---

**Configure Data Traffic Exchange across Private WANs**

When a vEdge router is connected to a private WAN, such as an MPLS or a metro Ethernet network, the carrier hosting the private network does not advertise the IP address of that vEdge router over the internet. (This IP address is associated with the TLOC on that vEdge router.) This means that remote vEdge routers are not able to learn how to reach that router and hence are not able to exchange data traffic with it directly over the private network.

To allow the vEdge router behind the private network to communicate directly over the private WAN with other vEdge routers, you direct the data traffic to a loopback interface rather than to the actual physical WAN interface. The overlay network can then advertise that the local router is reachable via its loopback address. To make it possible for the data traffic to actually be transmitted out the WAN interface, you bind the loopback interface to the physical WAN interface to the private network.

To configure VPN 0 so that it carries data traffic across private WANs:

1. Configure the loopback interface, assigning it an IP address:
   ```
   vEdge(config)# vpn 0 loopback number ip address prefix/length
   vEdge(config-loopback)# no shutdown
   ```
2. Configure the loopback interface to be a transport interface:
   ```
   vEdge(config-loopback)# tunnel-interface
   ```
3. Set the color of the loopback interface to be one of the primatel colors—**metro-ethernet, mpls**, and **private1** through **private6**. You must configure this same color on the loopback interfaces of all vEdge routers in the same private LAN.
   
   ```
   vEdge(config-tunnel-interface)# color color
   ```

   Use the `show interface` command to check that the loopback interface in configured properly, as a transport interface with the proper IP address and color.

   If a single vEdge router is connected to two (or more) different private networks, create a loopback interface for each private network, associate a carrier name with the interface so that the router can distinguish between the two private WANs, and "bind" the loopback interface to the physical interface that connects to the appropriate private WAN:

   1. Configure the loopback interface, assigning it an IP address:
      ```
      vEdge(config)# vpn 0 loopback number ip address prefix/length
      vEdge(config-loopback)# no shutdown
      ```
   2. Configure the loopback interface to be a transport interface and bind it to a physical interface:
      ```
      vEdge(config-loopback)# tunnel-interface bind ge slot/port
      ```
   3. Configure a carrier name and TLOC color on the loopback interface:
      ```
      vEdge(config-tunnel-interface)# carrier carrier-name
      vEdge(config-tunnel-interface)# color color
      ```
   4. On the physical interface, configure its IP address, and enable it:
      ```
      vEdge(config)# vpn 0 interface ge slot/port ip address prefix/length
      vEdge(config-ge)# no shutdown
      ```

---

**Configure the Management VPN (VPN 512)**

In the Viptela overlay network, VPN 512 is the network management VPN. It carries out-of-band management traffic in the overlay network. VPN 512 is configured and enabled by default on all Viptela devices. It contains the interface used for management traffic. For vEdge routers, this interface is generally a Gigabit Ethernet (**ge**) interface, and for other Viptela devices it is an **eth** interface. DHCP is enabled by default on the management interface. The default configuration for VPN 512 on a vEdge router looks like this:

```
vpn 512
  interface ge0/0
  ip dhcp-client
  no shutdown
```

VPN 512 must be present on all Viptela devices so that they are always reachable on the network. You can configure additional parameters for VPN 512 if you choose.

---

**Configure VPNs To Carry Data Traffic**

VPNs other than VPN 0 and VPN 512 are used to carry data traffic across the overlay network. These VPNs are sometimes referred to as **service-side VPNs**. For these VPNs to operate, each one must have an operational interface...
To create a data traffic VPN:

1. Configure the VPN:
   ```
   vEdge(config)# vpn number
   vEdge(config-vpn)#
   ```
   The VPN number can be in the range 1 through 511, and 513 through 65535.

2. Configure at least one interface in the VPN and its IP address:
   ```
   vEdge(config-vpn)# interface interface-name ip address address/prefix
   vEdge(config-interface)#
   ```
   The interface name has the format `ge(slot/port)`, where the slot is generally 0 through 7 (depending on the device) and the port is 0 through 8. If you are configuring VLANs, specify a subinterface name in the format `ge(slot/port).vlan`, where the VLAN number can be in the range 1 through 4094. (VLAN numbers 0 and 4095 are reserved.) The interface name can also be `gre(number)`, `ipsec(number)`, `loopback(string)`, `natpool(number)`, or `ppp(number)`.  

3. Activate the interface:
   ```
   vEdge(config-interface)# no shutdown
   ```

4. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from that VPN:
   ```
   vEdge(config-vpn)# dns ip-address
   ```

5. If desired, configure IPv4 static routes in the VPN:
   ```
   vEdge(config-vpn)# ip route prefix/length next-hop [administrative-distance]
   ```

6. Configure any other properties specific to the interface or to VPN.

7. Activate the configuration:
   ```
   vEdge(config)# commit
   ```

Here is an example of a configuration for VPN 1:

```
vpn 1
dns 1.2.3.4 primary
router ospf
   redistribute ospf 10 route-policy test-policy
   area 0
      interface ge0/3
      exit
      exit
!
interface ge0/3
ip address 10.10.10.1/24
no shutdown
!
```

**Dual-Stack Operation**

When a Viptela device establishes an IPsec tunnel for control traffic between a local TLOC and a remote TLOC, or when a device establishes a BFD tunnel for data plane traffic between a local and a remote TLOC, an IPv6 tunnel is established in the following situations:
• The local device has only an IPv6 address, and the remote device has an IPv6 address.

• The remote device has only an IPv6 address, and the local device has an IPv6 address.

If both the local and remote devices have IPv4 addresses, IPsec and BFD always establish an IPv4 tunnel.

Additional Information

Configuring Interfaces
Configuring Multicast Overlay Routing
Configuring Unicast Overlay Routing
Segmentation (VPN) Overview
Segmentation (VPN) Configuration Examples