Configuring an Application-Based Layer 3 VPN Topology

This example illustrates an application-based mechanism for forwarding traffic into a Layer 3 VPN. Typically, one or more interfaces are associated with, or bound to, a VPN by including them in the configuration of the VPN routing instance. By binding the interface to the VPN, the VPN's VRF table is used to make forwarding decisions for any incoming traffic on that interface. Binding the interface also includes the interface local routes in the VRF table, which provides next-hop resolution for VRF routes.

In this example, a firewall filter is used to define which incoming traffic on an interface is forwarded by means of the standard routing table, inet.0, and which incoming traffic is forwarded by means of the VRF table. You can expand this example such that incoming traffic on an interface can be redirected to one or more VPNs. For example, you can define a configuration to support a VPN that forwards traffic based on source address, that forwards Hypertext Transfer Protocol (HTTP) traffic, or that forwards only streaming media.

For this configuration to work, the following conditions must be true:

- The interfaces that use filter-based forwarding must not be bound to the VPN.
- Static routing must be used as the means of routing.
- You must define an interface routing table group that is shared among inet.0 and the VRF tables to provide local routes to the VRF table.

This example consists of two client hosts (Client D and Client E) that are in two different VPNs and that want to send traffic both within the VPN and to the Internet. The paths are defined as follows:

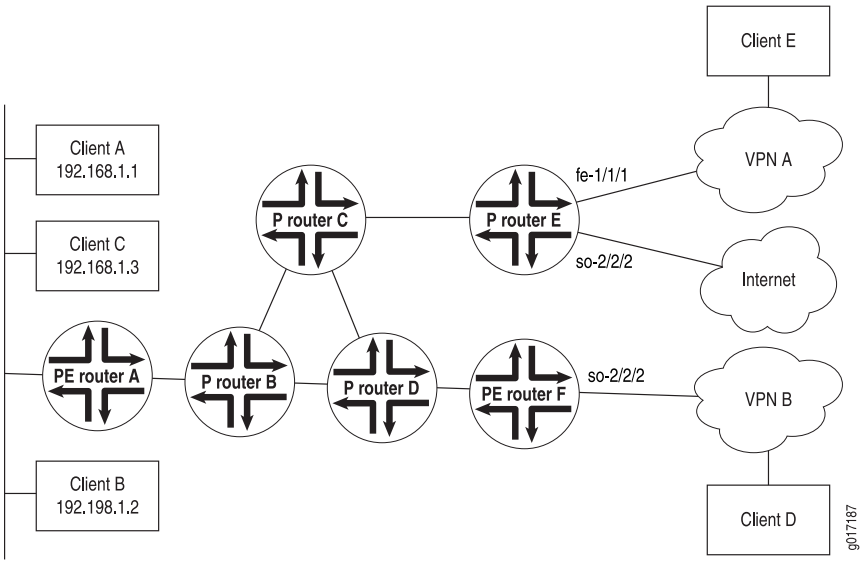
- Client A sends traffic to Client E over VPN A with a return path that also uses VPN A (using the VPN's VRF table).
- Client B sends traffic to Client D over VPN B with a return path that uses standard destination-based routing (using the inet.0 routing table).
- Clients B and C send traffic to the Internet using standard routing (using the inet.0 routing table), with a return path that also uses standard routing.

This example illustrates that there are a large variety of options in configuring an application-based Layer 3 VPN topology. This flexibility has application in many network implementations that require specific traffic to be forwarded in a constrained routing environment.

This configuration example shows only the portions of the configuration for the filter-based forwarding, routing instances, and policy. It does not illustrate how to configure a Layer 3 VPN.

Figure 1 illustrates the network topology used in this example.

Figure 1: Application-Based Layer 3 VPN Example Configuration



Configuration on Router A

On Router A, you configure the interface to Clients A, B, and C. The configuration evaluates incoming traffic to determine whether it is to be forwarded by means of VPN or standard destination-based routing.

First, you apply an inbound filter and configure the interface:

```
[edit]
interfaces {
   fe-1/1/0 {
      unit 0 {
         family inet {
            filter {
               input fbf-vrf;
            }
            address 192.168.1.1/24;
         }
      }
   }
}
```

Because the interfaces that use filter-based forwarding must not be bound to a VPN, you must configure an alternate method to provide next-hop routes to the VRF table. You do this by defining an interface routing table group and sharing this group among all the routing tables:

```
[edit]
```

```
routing-options {
   interface-routes {
      rib-group inet if-rib;
   }
   rib-groups {
      if-rib {
         import-rib [ inet.0 vpn-A.inet.0 vpn-B.inet.0 ];
      }
   }
}
```

You apply the following filter to incoming traffic on interface **fe-1/1/0.0**. The first term matches traffic from Client A and forwards it to the routing instance for VPN A. The second term matches traffic from Client B that is destined for Client D and forwards it to the routing instance for VPN B. The third term matches all other traffic, which is forwarded normally by means of destination-based forwarding according to the routes in **inet.0**.

```
[edit firewall family family-name]
filter fbf-vrf {
   term vpnA {
      from {
         source-address {
            192.168.1.1/32;
         }
      }
      then {
         routing-instance vpn-A;
      }
   }
   term vpnB {
      from {
         source-address {
            192.168.1.2/32;
         }
         destination-address {
            192.168.3.0/24;
         }
      }
      then routing-instance vpn-B;
   }
}
term internet {
   then accept;
}
```

You then configure the routing instances for VPN A and VPN B. Notice that these statements include all the required statements to define a Layer 3 VPN except for the **interface** statement.

```
[edit]
routing-instances {
   vpn-A {
      instance-type vrf;
      route-distinguisher 172.21.10.63:100;
```

```
        vrf-import vpn-A-import;
        vrf-export vpn-A-export;
    }
    vpn-B {
        instance-type vrf;
        route-distinguisher 172.21.10.63:200;
        vrf-import vpn-B-import;
        vrf-export vpn-B-export;
    }
}
```

Configuration on Router E

On Router E, configure a default route to reach the Internet. You should inject this route into the local IBGP mesh to provide an exit point from the network.

```
[edit]
routing-options {
    static {
        route 0.0.0.0/0 next-hop so-2/2/2.0 discard
    }
}
```

Configure the interface to Client E so that all incoming traffic on interface fe-1/1/1.0 that matches the VPN policy is forwarded over VPN A:

```
[edit]
routing-instances {
    vpn-A {
        interface fe-1/1/1.0
        instance-type vrf;
        route-distinguisher 172.21.10.62:100;
        vrf-import vpn-A-import;
        vrf-export vpn-A-export;
        routing-options {
            static {
                route 192.168.2.0/24 next-hop fe-1/1/1.0;
            }
        }
    }
}
```

Configuration on Router F

Again, because the interfaces that use filter-based forwarding must not be bound to a VPN, you configure an alternate method to provide next-hop routes to the VRF table by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP:

```
[edit]
routing-options {
    interface-routes {
        rib-group inet if-rib;
```

```
    }
    rib-groups {
        if-rib {
            import-rib [ inet.0 vpn-B.inet.0 ];
        }
    }
}
```

To direct traffic from VPN B to Client D, you configure the routing instance for VPN B on Router F. All incoming traffic from Client D on interface **so-3/3/3.0** is forwarded normally by means of the destination address based on the routes in **inet.0**.

```
[edit]
routing-instances {
    vpn-B {
        instance-type vrf;
        route-distinguisher 172.21.10.64:200;
        vrf-import vpn-B-import;
        vrf-export vpn-B-export;
        routing-options {
            static {
                route 192.168.3.0/24 next-hop so-3/3/3.0;
            }
        }
    }
}
```

Published: 2010-04-27