Network Virtualization—Path Isolation Design Guide

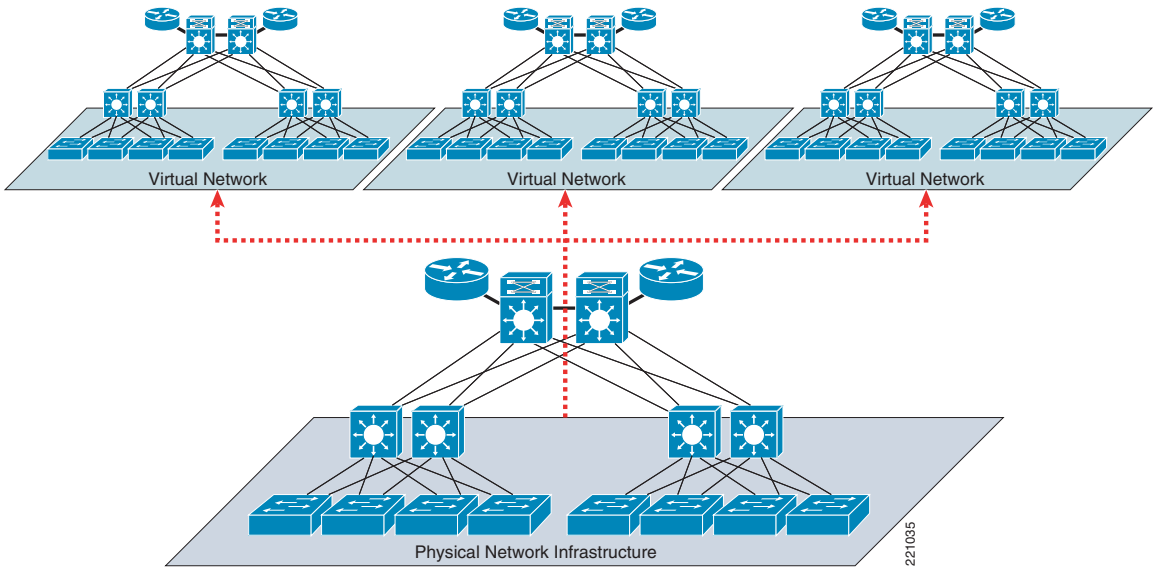
Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Introduction

The term *network virtualization* refers to the creation of logical isolated network partitions overlaid on top of a common enterprise physical network infrastructure, as shown in Figure 1.

Figure 1 Creation of Virtual Networks



Each partition is logically isolated from the others, and must provide the same services that are available in a traditional dedicated enterprise network. The end user experience should be as if connected to a dedicated network providing privacy, security, an independent set of policies, service level, and even routing decisions. At the same time, the network administrator can easily create and modify virtual work environments for various user groups, and adapt to changing business requirements adequately. The latter is possible because of the ability to create security zones that are governed by policies enforced centrally; these policies usually control (or restrict) the communication between separate virtual

networks or between each logical partition and resources that can be shared across virtual networks. Because policies are centrally enforced, adding or removing users and services to or from a VPN requires no policy reconfiguration. Meanwhile, new policies affecting an entire group can be deployed centrally at the VPN perimeter. Thus, virtualizing the enterprise network infrastructure provides the benefits of using multiple networks but not the associated costs, because operationally they should behave like one network (reducing the relative OPEX costs).

Network virtualization provides multiple solutions to business problems and drivers that range from simple to complex. Simple scenarios include enterprises that want to provide Internet access to visitors (guest access). The stringent requirement in this case is to allow visitors external Internet access, while simultaneously preventing any possibility of unauthorized connection to the enterprise internal resources and services. This can be achieved by dedicating a logical "virtual network" to handle the entire guest communication path. Internet access can also be combined with connectivity to a subset of the enterprise internal resources, as is typical in partner access deployments.

Another simple driver for network virtualization is the creation of a logical partition dedicated to the machines that have been quarantined as a result of a Network Admission Control (NAC) posture validation. In this case, it is essential to guarantee isolation of these devices in a remediation segment of the network, where only access to remediation servers is possible until the process of cleaning and patching the machine is successfully completed.

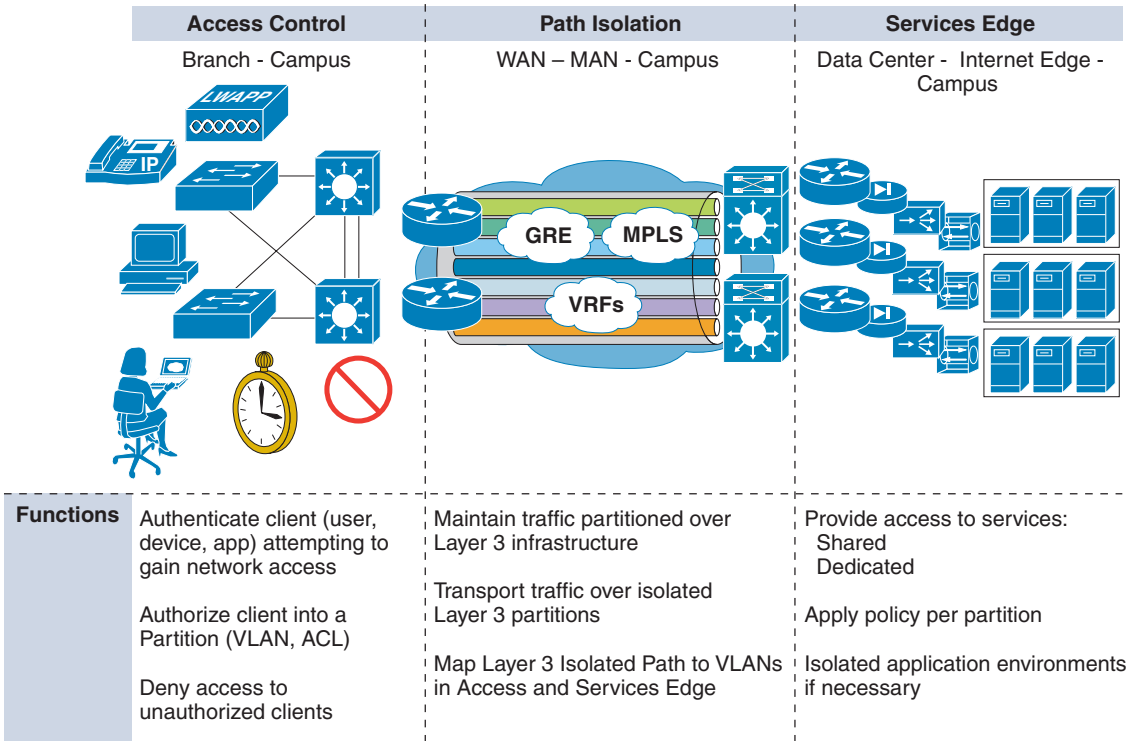
Complex scenarios include enterprise IT departments acting as a service provider, offering access to the enterprise network to many different "customers" that need logical isolation between them. In the future, users belonging to the same logical partitions will be able to communicate with each other and to share dedicated network resources. However, some direct inter-communication between groups may be prohibited. Typical deployment scenarios in this category include retail stores that provide on-location network access for kiosks or hotspot providers.

The architecture of an end-to-end network virtualization solution targeted to satisfy the requirements listed above can be separated in the following three logical functional areas:

- Access control
- Path isolation
- Services edge

Each area performs several functions and must interface with the other functional areas to provide the end-to-end solution (see Figure 2).

Figure 2 Network Virtualization Framework



The functionalities highlighted in Figure 2 are discussed in great detail in separate design guides, each one dedicated to a specific functional area.

- *Network Virtualization—Access Control Design Guide* (OL-13634-01)—Responsible for authenticating and authorizing entities connecting at the edge of the network; this allows assigning them to their specific network "segment", which usually corresponds to deploying them in a dedicated VLAN.
- *Network Virtualization—Services Edge Design Guide* (OL-13637-01)—Central policy enforcement point where it is possible to control/restrict communications between separate logical partitions or access to services that can be dedicated or shared between virtual networks.

The path isolation functional area is the focus of this guide.

This guide mainly discusses two approaches for achieving virtualization of the routed portion of the network:

- Policy-based network virtualization—Restricts the forwarding of traffic to specific destinations, based on a policy, and independently from the information provided by the control plane. A classic example of this uses ACLs to restrict the valid destination addresses to subnets in the VPN.
- Control plane-based network virtualization—Restricts the propagation of routing information so that only subnets that belong to a virtual network (VPN) are included in any VPN-specific routing tables and updates. This second approach is the main core of this guide, because it allows overcoming many of the limitations of the policy-based method.

Various path isolation alternatives technologies are discussed in the sections of this guide; for the reader to make good use of this guide, it is important to underline two important points:

- This guide discusses the implementation details of each path isolation technology to solve the business problems previously discussed, but is not intended to provide a complete description of each technology. Thus, some background reading is needed to acquire complete familiarity with

each topic. For example, when discussing MPLS VPN deployments, some background knowledge of the technology is required, because the focus of the document is discussing the impact of implementing MPLS VPN in an enterprise environment, and not its basic functionality.

- Not all the technologies found in this design guide represent the right fit for each business requirement. For example, the use of distributed access control lists (ACLs) or generic routing encapsulation (GRE) tunnels may be particularly relevant in guest and partner access scenarios, but not in deployments aiming to fulfill different business requirements. To properly map the technologies discussed here with each specific business requirement, see the following accompanying deployment guides:
  - *Network Virtualization—Guest and Partner Access Deployment Guide* (OL-13635-01)
  - *Network Virtualization—Network Admission Control Deployment Guide* (OL-13635-01)

Path Isolation Overview

Path isolation refers to the creation of independent logical traffic paths over a shared physical network infrastructure. This involves the creation of VPNs with various mechanisms as well as the mapping between various VPN technologies, Layer 2 segments, and transport circuits to provide end-to-end isolated connectivity between various groups of users.

The main goal when segmenting the network is to preserve and in many cases improve scalability, resiliency, and security services available in a non-segmented network. Any technology used to achieve virtualization must also provide the necessary mechanisms to preserve resiliency and scalability, and to improve security.

A hierarchical IP network is a combination of Layer 3 (routed) and Layer 2 (switched) domains. Both types of domains must be virtualized and the virtual domains must be mapped to each other to keep traffic segmented. This can be achieved when combining the virtualization of the network devices (also referred to as “device virtualization”) with the virtualization of their interconnections (known as “data path virtualization”).

In traditional (that is, not virtualized) deployments, high availability and scalability are achieved through a hierarchical and modular design based on the use of three layers: access, distribution, and core.

Note For more information on the recommended design choices to achieve high availability and scalability in campus networks, see the following URL: http://www.cisco.com/en/US/netsol/ns656/networking_solutions_design_guidances_list.html#anchor2

Much of the hierarchy and modularity discussed in the documents referenced above rely on the use of a routed core. Nevertheless, some areas of the network continue to benefit from the use of Layer 2 technologies such as VLANs (typically in a campus environment) and ATM or Frame Relay circuits (over the WAN). Thus, a hierarchical IP network is a combination of Layer 3 (routed) and Layer 2 (switched) domains. Both types of domains must be virtualized and the virtual domains must be mapped to each other to keep traffic segmented.

Virtualization in the Layer 2 domain is not a new concept: VLANs have been used for years. What is now required is a mechanism that allows the extension of the logical isolation over the routed portion of the network. Path isolation is the generic term referring to this logical virtualization of the transport. This can be achieved in various ways, as is discussed in great detail in the rest of this guide.

Virtualization of the transport must address the virtualization of the network devices as well as their interconnection. Thus, the virtualization of the transport involves the following two areas of focus:

- Device virtualization—The virtualization of the network device; this includes all processes, databases, tables, and interfaces within the device.
- Data path virtualization—The virtualization of the interconnection between devices. This can be a single-hop or multi-hop interconnection. For example, an Ethernet link between two switches provides a single-hop interconnection that can be virtualized by means of 802.1q VLAN tags; whereas for Frame Relay or ATM transports, separate virtual circuits can be used to provide data path virtualization. When an IP cloud is separating two virtualized devices, a multi-hop interconnection is required to provide end-to-end logical isolation. An example of this is the use of tunnel technologies (for example, GRE) established between the virtualized devices deployed at the edge of the network.

In addition, within each networking device there are two planes to virtualize:

- Control plane—All the protocols, databases, and tables necessary to make forwarding decisions and maintain a functional network topology free of loops or unintended black holes. This plane can be said to draw a clear picture of the topology for the network device. A virtualized device must have a unique picture of each virtual network it handles; thus, there is the requirement to virtualize the control plane components.
- Forwarding plane—All the processes and tables used to actually forward traffic. The forwarding plane builds forwarding tables based on the information provided by the control plane. Similar to the control plane, each virtual network has a unique forwarding table that needs to be virtualized.

Furthermore, the control and forwarding planes can be virtualized at different levels, which map directly to different layers of the OSI model. For instance, a device can be VLAN-aware and therefore be virtualized at Layer 2, yet have a single routing table, which means it is not virtualized at Layer 3. The various levels of virtualization are useful, depending on the technical requirements of the deployment. There are cases in which Layer 2 virtualization is enough, such as a wiring closet. In other cases, virtualization of other layers may be necessary; for example, providing virtual firewall services requires Layer 2, 3, and 4 virtualization, plus the ability to define independent services on each virtual firewall, which perhaps is Layer 7 virtualization.

Policy-Based Path Isolation

Policy-based path isolation techniques restrict the forwarding of traffic to specific destinations, based on a policy and independently of the information provided by the forwarding control plane. A classic example of this uses an ACL to restrict the valid destination addresses to subnets that are part of the same VPN.

Policy-based segmentation is limited by two main factors:

- Policies must be configured pervasively (that is, at every edge device representing the first L3 hop in the network)
- Locally significant information (that is, IP address) is used for policy selection

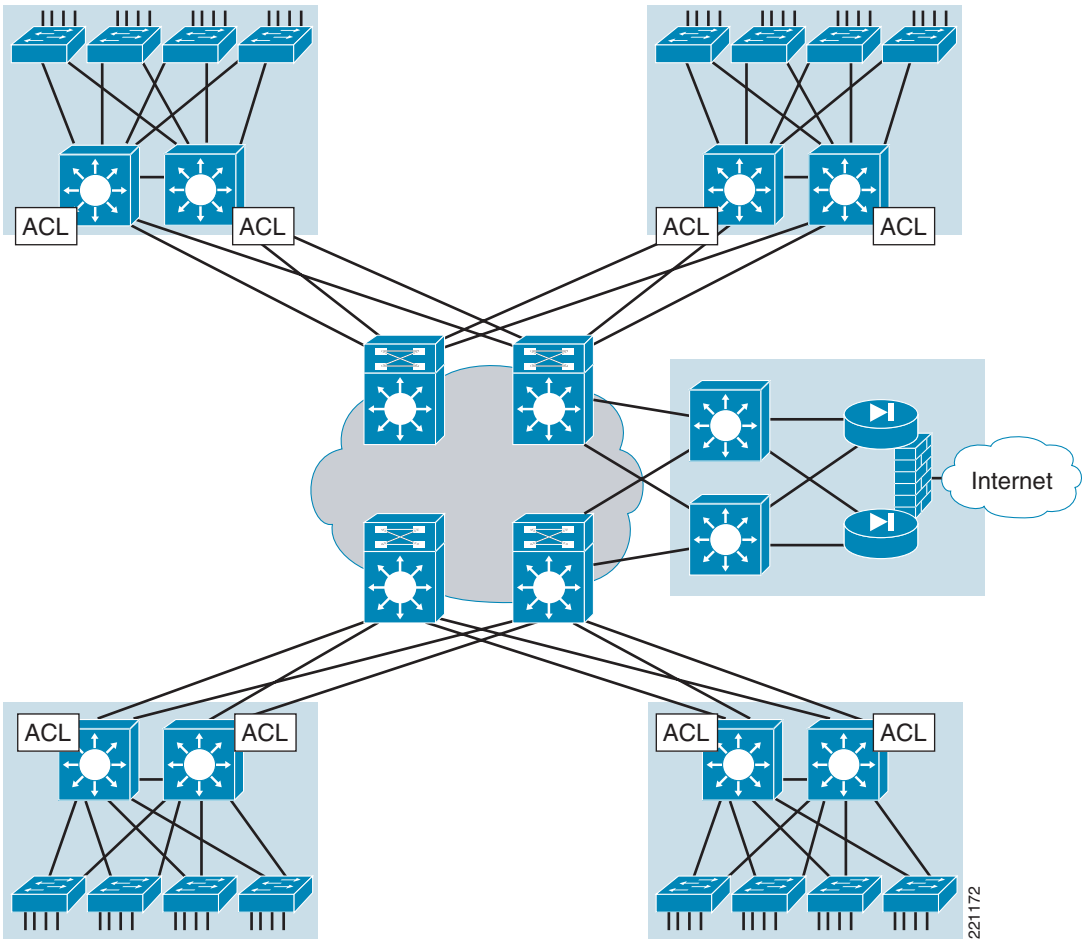
The configuration of distributed policies can be a significant administrative burden, is error prone, and causes any update in the policy to have widespread impact.

Because of the diverse nature of IP addresses, and because policies must be configured pervasively, building policies based on IP addresses does not scale very well. Thus, IP-based policy-based segmentation has limited applicability.

As discussed subsequently in Path Isolation Using Distributed Access Control Lists, page 14, using policy-based path isolation with the tools available today (ACLs) is still feasible for the creation of virtual networks with many-to-one connectivity requirements, but it is very difficult to provide any-to-any connectivity with such technology For example, hub-and-spoke topologies are required to

provide an answer to the guest access problem, where all the visitors need to have access to a single resource (the Internet). Using ACLs in this case is still manageable because the policies are identical everywhere in the network (that is, allow Internet access, deny all internal access). The policies are usually applied at the edge of the Layer 3 domain. Figure 3 shows ACL policies applied at the distribution layer to segment a campus network.

Figure 3 Policy-Based Path Isolation with Distributed ACLs



Control Plane-Based Path Isolation

Control plane-based path isolation techniques restrict the propagation of routing information so that only subnets that belong to a virtual network (VPN) are included in any VPN-specific routing tables and updates. To achieve control plane virtualization, a device must have many control/forwarding instances, one for each VPN. This is possible when using the virtual routing and forwarding (VRF) technology that allows for the virtualization of the L3 devices.

Network Device Virtualization with VRF

A VRF instance consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. As shown in Figure 4, the use of VRF technology allows the customer to virtualize a network device from a Layer 3 standpoint, creating different "virtual routers" in the same physical device.

Note A VRF is not strictly a virtual router because it does not have dedicated memory, processing, or I/O resources, but this analogy is helpful in the context of this guide.

Figure 4 Virtualization of a Layer 3 Network Device

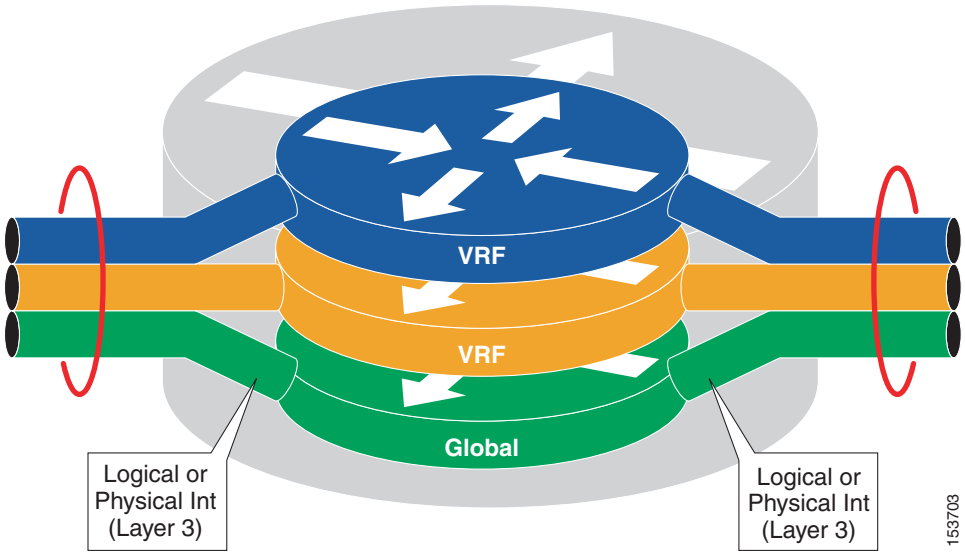


Table 1 provides a listing of the VRF-lite support on the various Cisco Catalyst platforms that are typically found in an enterprise campus network. As is clarified in following sections, VRF-lite and MPLS support are different capabilities that can be used to provide separate path isolation mechanisms (VRF-lite + GRE, MPLS VPN, and so on.)

Table 1 VRF-Lite Support on Cisco Catalyst Switches

Platform	Minimum Software Release	Number of VRF	VRF Routing Protection Support	Full MPLS Support
Catalyst 3550	12.1(11)EA1 (EMI resp. IP Svc.)	7[1]	Yes	No
Catalyst 3560	12.2(25)SEC (min. IP Svc.)	26[1]	Yes	No
Catalyst 3750	12.2(25)SEC (min. IP Svc.)	26[1]	Yes	No
Catalyst 3750 Metro	12.1(14)AX (min. IP Svc.)	26	Yes	Yes (min. Adv. IP Svc.)

Table 1 VRF-Lite Support on Cisco Catalyst Switches (continued)

Platform	Minimum Software Release	Number of VRF	VRF Routing Protection Support	Full MPLS Support
Catalyst 4500-SupIII/IV/V/V-10GE	12.2(18)EW[2]	64[1]	Yes	No
Catalyst 4948/4948-10GE	12.2(20)EWA[2]	64[1]	Yes	No
Catalyst ME-X4924-10GE	12.2(31)SGA	64[1]	Yes	No
Catalyst 6500/7600-Sup720 (PFC3A)	12.2(17b)SXA	1000	Yes	No!
Catalyst 6500/7600-Sup720-3B	12.2(18)SXD	1000	Yes	Yes (min. Adv. IP Svc.)
Catalyst 6500/7600-Sup720-3BXL	12.2(17b)SXA	1000	Yes	Yes (min. Adv. IP Svc.)
Catalyst 6500/7600-Sup32	12.2(18)SXF	1000	Yes	Yes (min. Adv. IP Svc.)
Catalyst ME-C6524 (currently DC only)	12.2(18)ZU	1000	Yes	Yes (min. Adv. IP Svc.)

1. No multicast support within VRFs
2. Starting with 12.2(25)SG, VRF-lite is *only* supported in Enhanced Service Image -> SupII+ no longer provides VRFs.

One important thing to consider with regard to the information above is that a Catalyst 6500 equipped with Supervisor 2 is capable of supporting VRFs only when using optical switching modules (OSMs). The OSM implementation is considered legacy and more applicable to a WAN environment. As a consequence, a solution based on VRF should be taken into consideration in a campus environment only if Catalyst 6500 platforms are equipped with Supervisors 32 or 720 (this is why this option is not displayed in Table 1).

The use of Cisco VRF-Lite technology has the following advantages:

- Allows for true routing and forwarding separation—Dedicated data and control planes are defined to handle traffic belonging to groups with various requirements or policies. This represents an additional level of segregation and security, because no communication between devices belonging to different VRFs is allowed unless explicitly configured.
- Simplifies the management and troubleshooting of the traffic belonging to the specific VRF, because separate forwarding tables are used to switch that traffic—These data structures are different from the one associated to the global routing table. This also guarantees that configuring the overlay network does not cause issues (such as routing loops) in the global table.
- Enables the support for alternate default routes—The advantage of using a separate control and data plane is that it allows for defining a separate default route for each virtual network (VRF). This can be useful, for example, in providing guest access in a deployment when there is a requirement to use the default route in the global routing table just to create a black hole for unknown addresses to aid in detecting certain types of worm and network scanning attacks.

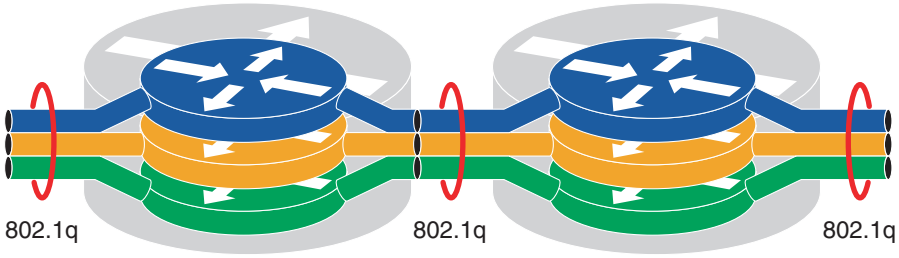
  In this example, employee connectivity to the Internet is usually achieved by using a web proxy device, which can require a specific browser configuration on all the machines attempting to connect to the Internet or having the need to provide valid credentials. Although support for web proxy servers on employee desktops is common practice, it is not desirable to have to reconfigure a guest browser to point to the proxy servers. As a result, the customer can configure a separate forwarding table for using an alternative default route in the context of a VRF, to be used exclusively for a specific type of traffic, such as guest traffic. In this case, the default browser configuration can be used.

Data Path Virtualization—Single- and Multi-Hop Techniques

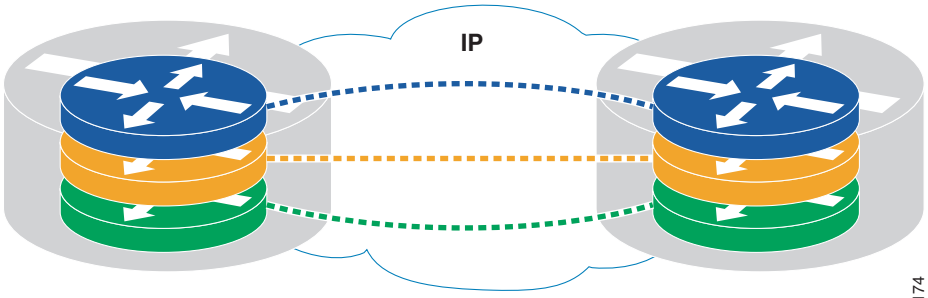
The VRF achieves the virtualization of the networking devices at Layer 3. When the devices are virtualized, the virtual instances in the various devices must be interconnected to form a VPN. Thus, a VPN is a group of interconnected VRFs. In theory, this interconnection can be achieved by using dedicated physical links for each VPN (a group of interconnected VRFs). In practice, this is very inefficient and costly. Thus, it is necessary to virtualize the data path between the VRFs to provide logical interconnectivity between the VRFs that participate in a VPN.

The type of data path virtualization varies depending on how far the VRFs are from each other. If the virtualized devices are directly connected to each other (single hop), link or circuit virtualization is necessary. If the virtualized devices are connected through multiple hops over an IP network, a tunneling mechanism is necessary. Figure 5 illustrates single-hop and multi-hop data path virtualization.

Figure 5 Single- and Multi-Hop Data Path Virtualization



L2 based labeling allows single hop data path virtualization

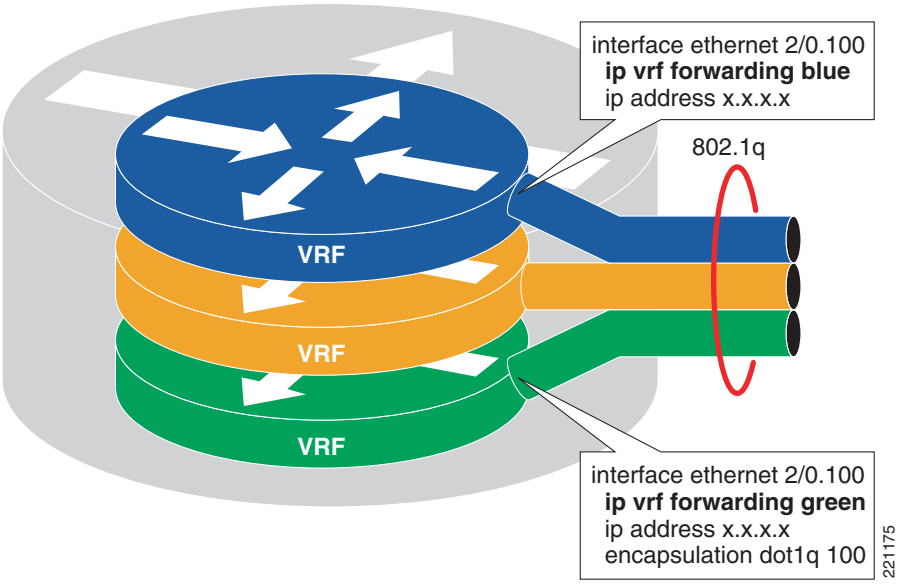


Tunnels allow multi-hop data path virtualization

The many technologies that virtualize the data path and interconnect VRFs are discussed in the next sections. The various technologies have benefits and limitations depending on the type of connectivity and services required. For instance, some technologies are very good at providing hub-and-spoke connectivity, while others provide any-to-any connectivity. The support for encryption, multicast, and other services also determine the choice of technologies to be used for the virtualization of the transport.

The VRFs must also be mapped to the appropriate VLANs at the edge of the network. This mapping provides continuous virtualization across the Layer 2 and Layer 3 portions of the network. The mapping of VLANs to VRFs is as simple as placing the corresponding VLAN interface at the distribution switch into the appropriate VRF. The same type of mapping mechanism applies to Layer 2 virtual circuits (ATM, Frame Relay) or IP tunnels that are handled by the router as a logical interface. The mapping of VLAN logical interfaces (Switch Virtual Interface [SVI]) and of sub-interfaces to VRFs is shown in Figure 6.

Figure 6 VLAN to VRF Mapping



Path Isolation Initial Design Considerations

Before discussing the various path isolation alternatives in more detail, it is important to highlight some initial considerations that affect the overall design presented in the rest of this guide. These assumptions are influenced by several factors, including the current status of the technology and the specific business requirements driving each specific solution. As such, they may change or evolve in the future; this guide will be accordingly updated to reflect this fact.

- Use of virtual networks for specific applications

  The first basic assumption is that even in a virtualized network environment, the global table is where most of the enterprise traffic is still handled. This means that logical partitions (virtual networks) are created to provide response to specific business problems (as, for example, guest Internet access), and users/entities are removed from the global table and assigned to these partitions only when meeting specific requirements (as, for example, being a guest and not an internal enterprise employee). The routing protocol traditionally used to provide connectivity to the various enterprise entities in global table (IGP) is still used for that purpose. In addition, the global IGP may also be used to provide the basic IP connectivity allowing for the creation of the logical overlay partitions; this is, for example, the case when implementing tunneling technologies such as VRF-Lite and GRE or MPLS VPN. In summary, the idea is to maintain the original global table design and "pull out" entities from the global table only for satisfying specific requirements (the business drivers previously discussed). This strategy allows support for gradual evolution to a virtualized from a non-virtualized network; also, it reduces the risk to existing production applications.

- Integration of VoIP technologies in a virtualized network

When deploying a VoIP architecture to be integrated in a virtualized network, the current best practice design recommends to keep the main components of the voice infrastructure (VoIP handsets, Cisco CallManagers, Cisco Unity Servers, and so on) in the global table, together with all the users that use voice services (using Cisco Communicator software, VT Advantage, and so on). Reasons for following this recommendation in this phase of the technology include the following:

  - Current lack of VRF-aware voice services such as Survivable Remote Site Telephony (SRST) or Resource Reservation Protocol (RSVP) for Call Admission Control (CAC), which would prevent a successful deployment of VoIP technologies at remote locations (without the burden of replicating the physical network infrastructure, which is against one of the main drivers for virtualizing the network). Also, Cisco CallManager does not currently officially support multi-tenant environments.

- Complex configuration required at the services edge of the network to allow the establishment of voice flows between entities belonging to separate VPNs. This would also require "punching" holes in the firewall deployed in this area of the network, increasing the security concerns of the overall solution.

- VoIP can be secured without requiring the creation of a dedicated logical partition for the voice infrastructure. There are proven tools and design recommendations that can be used for hardening the voice systems that are inherent in the system and do not require any form of network virtualization to be implemented. For more information, see the Voice SRND at the following URL: http://www.cisco.com/en/US/netsol/ns656/networking_solutions_design_guidances_list.html#anchor10

When the VoIP infrastructure is deployed in the global table, the direct consequence is the recommendation of keeping all the internal users that make use of VoIP applications (such as Cisco Communicator clients, for example) in the same domain, to not complicate the design too much when there is a need to establish voice flows between these users and, for example, the VoIP handsets. This is inline with the recommendation given in the first bullet point dictating the creation of virtual networks only for specific purposes.

- Deployment of network virtualization as an overlay design

  Another important initial assumption is that the deployment of a virtualized infrastructure constitutes an overlay design rather than a "rip-and-replace" approach. This means that the goal is the deployment of network virtualization without impacting (or just with limited impact to) network design that customers may already have in place. For example, if routing is already deployed using a specific IGP, the design should focus on demonstrating how to add services to that specific environment, rather than suggesting to tear apart the network and put a new network in place. This guide is focused on networks characterized by a single autonomous system (AS) and a single IGP-based environment, rather than large backbones with dual-redundant BGP cores.

- Security and VRF considerations

  Consider the following with regard to security and VRF:

  - A VRF-enabled network device is different from a completely virtualized device. The latter is usually referred to as "logical router", whereas the first is called "virtual router". A VRF-enabled device shares device resources (such as CPU, memory, hardware, and so on) between the various virtual instances supported. This essentially means that a failure of a problem with one of these shared elements affects all the virtual routers defined in the box.

  - In terms of isolation versus privacy, configuring separate VRFs allows support for multiple address spaces and for virtualizing both the control and data planes. However, simply doing this does not ensure the privacy of the information that is exchanged in the context of each VPN. To provide this extra layer of security, other technologies (such as IPsec) should be coupled with the specific path isolation strategy implemented.

  - The use of VRF does not eliminate the need for edge security features. As previously discussed, VRFs are enabled on the first L3 hop device; therefore, many of the security features that are recommended at the edge of the network (access layer) should still be implemented. This is true for identity-based techniques, such as 802.1x and MAB, which are discussed in *Network Virtualization—Access Control Design Guide* (OL-13634-01).

    However, it is important to highlight the requirement for integrating other security components, such as Catalyst Integrated Security Features (CISF) including DHCP Snooping, IP Source Guard, Dynamic ARP Inspection, or Port Security. In addition to these, Control Plane Policing (CPP) also needs to be considered to protect the CPU of the network devices. Another factor is that, as explained in the previous point above, a problem in a specific VRF may affect the CPU of the virtualized devices causing outages also in the other VRFs defined in the network device.

- QoS and network virtualization

  QoS and network virtualization are orthogonal problems in this phase of the technology. The main reason is that the DiffServ architecture has been deployed to be oriented around applications. Traffic originated by different applications (such as voice and video) is classified and marked at the edge of the network, and this marking information is used across the network to provide it with an appropriate level of service.

  In this phase of the technology, most enterprise routers and switches lack a virtual QoS mechanism. This means, for example, that the various input and output queues available on the network devices are not VRF-aware, which essentially implies that there is no capability to treat differently traffic originated by the same type of application in two different VPNs. For this reason, when discussing the deployment of QoS technologies in a virtualized network, there are two main strategies that can be adopted and that are applied to the various path isolation alternatives discussed in this paper:

  - Conform with the DiffServ standard functionality and keep classifying the traffic at the edge on an application base. This means that flows originating from the same application in different VPNs are treated in the same way across the network.
  - Define per-VPN policies. This means that all the traffic originating in a specific VPN is classified in the same way, independently from the application that originated it. This may find applicability for example in guest access scenarios, where the recommended strategy is to classify all the traffic originated from the guest user as best effort when below a predefined threshold. Traffic exceeding the threshold could for example be classified as scavenger so that it is the first to be dropped in case of network congestion.

The following sections provide more details on various path isolation techniques. The first is the use of distributed ACLs that, as previously mentioned, can be considered a policy-based mechanism, and is here discussed as a “legacy” way of limiting communication between users belonging to different network partitions. Various control plan-based techniques are then analyzed: first the use of VRF-Lite in conjunction with GRE tunneling, specifically recommended for deployments where an hub-and-spoke type of connectivity must be provided. For scenarios requiring any-to-any connectivity, the use of MPLS VPNs is discussed, highlighting the main differences between the enterprise deployments versus the more traditional service provider deployment.

Path Isolation Using Distributed Access Control Lists

The use of distributed ACLs represents a classic example of a policy-based path isolation mechanism to restrict the forwarding of traffic to specific destinations, based on a policy and independently of the information provided by the control plane. This allows restricting the group of valid destination addresses to the subnets that are configured as part of the same VPN (or virtual network).

Connectivity Requirements

The use of static ACLs at the edge of the network is the quickest way to provide traffic isolation, controlling and restricting communications between the various user groups. Most customers are comfortable with the use of ACLs to enforce security policies.

At the same time, using ACLs is recommended only in very specific scenarios where the network connectivity requirements are hub-and-spoke (multi-to-one). The main limitation of the ACL approach is the lack of scalability. The complexity of each distributed ACL is directly related to two main factors:

- The number of user groups that need to be supported
- Connectivity requirements between user groups

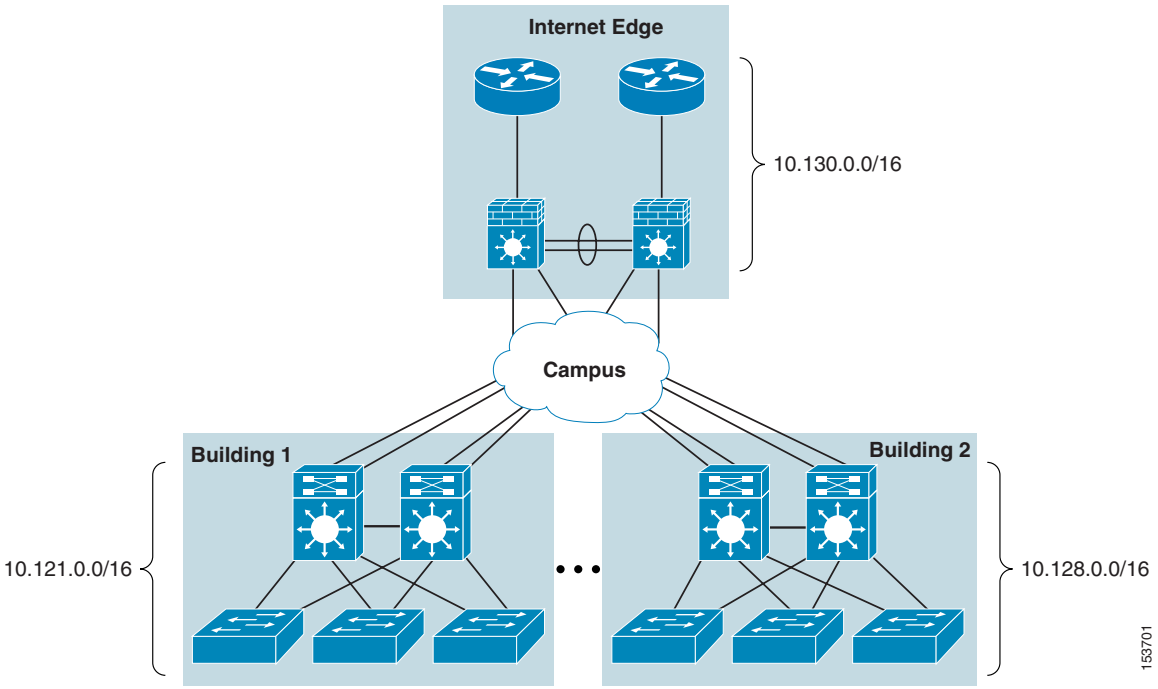
Defining ACLs in scenarios with a large number of groups requiring any-to-any connectivity can quickly become cumbersome from a management point of view. The goal is to propose this approach when the connectivity requirement is hub-and-spoke, so that it is possible to create a portable ACL template to be used across different spoke devices. Two typical applications that require this type of connectivity are guest access (where the target is providing access to the Internet as a centralized resource), and Network Admission Control (NAC) remediation (where connectivity must be restricted between unhealthy endpoints and a centralized remediation server). The common characteristic for these applications is the very limited number of user groups required (two in both cases), which makes the ACL approach a feasible technical candidate.

Configuration Details

The main goal is to create a generic ACL template that can be seamlessly used on all the required edge devices. This approach minimizes configuration and management efforts, and enhances the scalability of the overall solution. The same generic ACL should also be applied for both wired and wireless deployments. The specific wireless solution in place should affect the network device where the policy is applied, but not the format of the ACL itself.

Using ACLs to logically isolate traffic for specific categories of users (for example, employees and guests) on the same physical network implies that the control and data plan of the network needs to be shared between these different groups. The most immediate consequence is a limited freedom in assigning IP addresses to the various categories of users. The root of this problem is shown in Figure 7, which represents a generic campus network. This example refers to a guest access deployment where the hub devices are located in the Internet edge, but it can also be generic.

Figure 7 IP Addressing in the Campus Network



As shown in Figure 7, the recommended campus design dictates the assignment of IP addresses to various campus buildings in such a way that a summary route can be sent to the core (independent of the specific routing protocol being used). This isolates the buildings from a routing control point of view, contributing to the overall scalability and stability of the design. For example, 10.121.0.0/16 is the summary sent toward the core by the distribution layer devices belonging to Building 1.

Note The IP addresses used in this example simplify the description and are not intended to represent a best practice summarization schema.

As a result, all the IP subnets defined in each specific building block should be part of the advertised summary. This implies that subnets associated to the same user group but defined in separate buildings are part of different class B subnets. This clearly poses a challenge in defining a generic ACL template to be applied to devices belonging to different campus building blocks. The best way to achieve this is to define the edge policies without including the subnets from which the traffic is originated.

The recommended design described in this guide is based on the use of router ACLs (RACLs), which must be applied to Layer 3 interfaces. This means that in the multilayer campus design, the RACLs are applied to the distribution layer devices (representing the demarcation between Layer 2 and Layer 3 domains). The format of these ACLs remains the same, even in campus routed access deployments where the demarcation between Layer 2 and Layer 3 is pushed down to the access layer. The only difference is that, in this case, the RACLs need to be applied on the switched virtual interface (SVI) defined on the access layer devices.

RACLs are supported in hardware on Cisco Catalyst 6500 and 4500 platforms, which represent the devices most commonly deployed in the distribution layer of each campus building block. For more information, see the following URLs:

- http://www.cisco.com/en/US/customer/products/hw/switches/ps708/products_white_paper09186a00800c9470.shtml

- http://www.cisco.com/en/US/partner/products/hw/switches/ps663/products_tech_note09186a008054a499.shtml

The simplest RACL that can be deployed for a generic hub-and-spoke scenario is as follows:

```
ip access-list extended SEGM-RACL
 10 permit udp any any eq bootps
 20 permit udp any host <DNS-Server-IP> eq domain
 30 deny ip any <protected_prefixes>
 40 permit ip any <target_prefixes>
```

- Statements 10 and 20 allow connectivity to receive DHCP and DNS services (if needed).
- Statement 30 denies connectivity to protected resources that should not being accessed from this specific category of users.
- Statement 40 restricts connectivity only to the subset of required prefixes. The list of required prefixes varies, depending on the specific application. For example, in the case of guest access, it might be all the public IP addresses representing the Internet; for NAC remediation, it might be represented by the remediation server.

Note As previously mentioned, this ACL is generic enough to be applied to various edge devices. The key to doing this is to avoid the use of the source IP address in ACL statements.

RACLs derive their name from the fact that they need to be applied on Layer 3 (routed) interfaces. The Layer 3 interface where the RACL is applied depends on the specific type of network access used. For wired clients, the Layer 3 interfaces are the SVI (VLAN interface) defined on the distribution layer device (traditional design) or on the access layer devices (routed access design). The configuration for a generic SVI is as follows:

```
interface Vlan50
 description Wired-client-floor1
 ip address 10.124.50.2 255.255.255.0
 ip access-group SEGM-RACL in
```

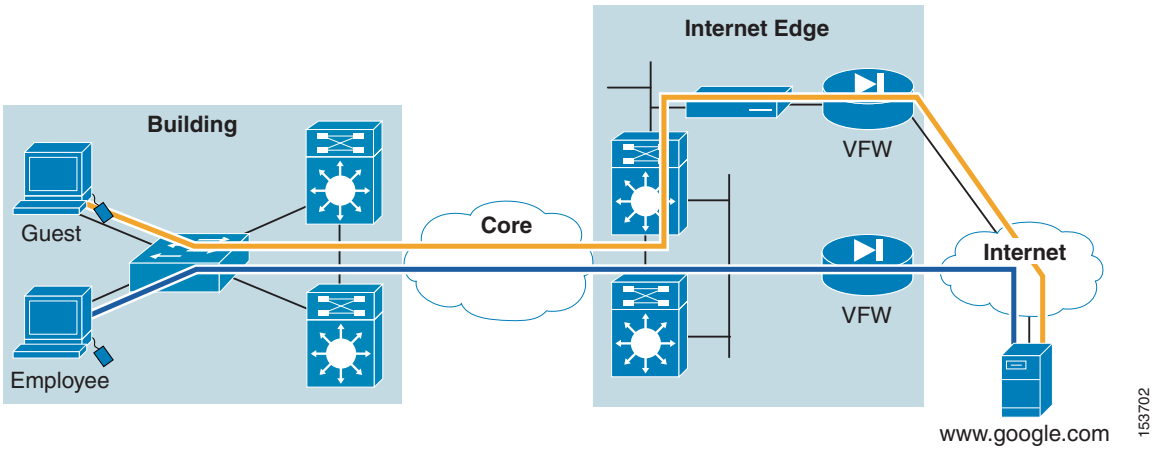
For wireless clients, it depends on the specific deployment in place. For traditional Cisco Aironet deployments and deployments using WLAN controllers, the situation is very similar to the wired case, and the ACL is applied on the SVIs defined on the distribution or access layer devices. For WLSM designs, where all the data traffic is tunneled from each distributed access point to a centralized Catalyst 6500 equipped with WLSM, the RACL can be directly applied on the receiving multipoint GRE (mGRE) interfaces defined on this centralized device, as follows:

```
interface Tunnel160
 description mGRE for clients-floor1
 ip address 10.121.160.1 255.255.255.0
 ip access-group SEGM-RACL in
```

Path Differentiation

Another aspect to consider is the problem of path differentiation. In some scenarios, you might need to redirect the traffic to a specific direction when it gets to the hub device. For example, this can be relevant in a guest access scenario where traffic might need to be enforced through a web authentication appliance. The solution uses policy-based routing (PBR). The following configuration samples and considerations refer to a guest access application, but their validity can easily be extended to other applications. Without going into specific detail on the problems associated with web authentication, note that web authentication appliances are usually deployed in-band, so you must devise a way to enforce the guest traffic through them, as illustrated in Figure 8.

Figure 8 Traffic Flows for Various Categories of Users



An internal employee and a guest pointing to the same final destination (in this example, www.google.com) must take two different paths. The employee can connect directly to the Internet after going through a firewall (or a firewall context, as shown in Figure 8). The guest must first be forced through the web authentication appliance to complete an authentication process. The recommended way to accomplish this is by using PBR on the network devices in the Internet edge, connecting to the campus core (two Catalyst 6500s in this example).

Note On Catalyst 6500 platforms using Supervisor 2 with PFC2 or Supervisor 720 with PFC3, PBR is fully supported in hardware using a combination of security and the ACL ternary content addressable memory (TCAM) feature, and the hardware adjacency table. Although a detailed description of PBR is beyond the scope of this guide, note that PBR does consume ACL TCAM resources and adjacency table entries. In Supervisor 2 with PFC2, 1024 of the 256 K available hardware adjacencies are reserved for PBR. In Supervisor 720 with PFC3, 2048 of the one million available hardware adjacencies are reserved for PBR.

The considerations about the IP range assignment to the guest subnets made in the previous section also have an impact on the configuration of the ACL to be used for policing the traffic in the Internet edge. It is unlikely that you can summarize all the guest subnets in a limited number of statements. More likely, a separate ACL statement needs to be added for each specific guest subnet defined in each campus building block, as shown in the following configuration sample:

```
ip access-list extended TO-WEB-AUTH-DEVICE
 permit ip 10.121.150.0 0.0.0.255 any
 permit ip 10.121.160.0 0.0.0.255 any
 permit ip 10.122.150.0 0.0.0.255 any
 ..........................................................................
 permit ip 10.128.160.0 0.0.0.255 any
!
route-map guest-to-WEB-AUTH-DEVICE permit 10
 match ip address TO-WEB-AUTH-DEVICE
 set ip next-hop 172.18.3.30
```

Note The address specified in the **set ip next-hop** statement is the internal interface of the web authentication appliance.

The route map must then be applied on all the physical interfaces connecting the Internet edge devices to the core of the network, as follows:

```
interface TenGigabitEthernet3/1
 description 10GigE link to Core Switch 1
 ip address 10.122.0.7 255.255.255.254
 ip policy route-map guest-to-WEB-AUTH-DEVICE
```

High Availability Considerations

The resiliency of a solution based on the use of distributed ACLs is achieved by implementing the recommended campus design. More information on this subject is beyond the scope of this guide. For more information, see the campus HA documents at the following URLs:

- http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/cdccont_0900aecd801a8a2d.pdf
- http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/cdccont_0900aecd801a89fc.pdf

Challenges and Limitations of Distributed ACLs

Some of the challenges and limitations of the distributed ACL approach are as follows:

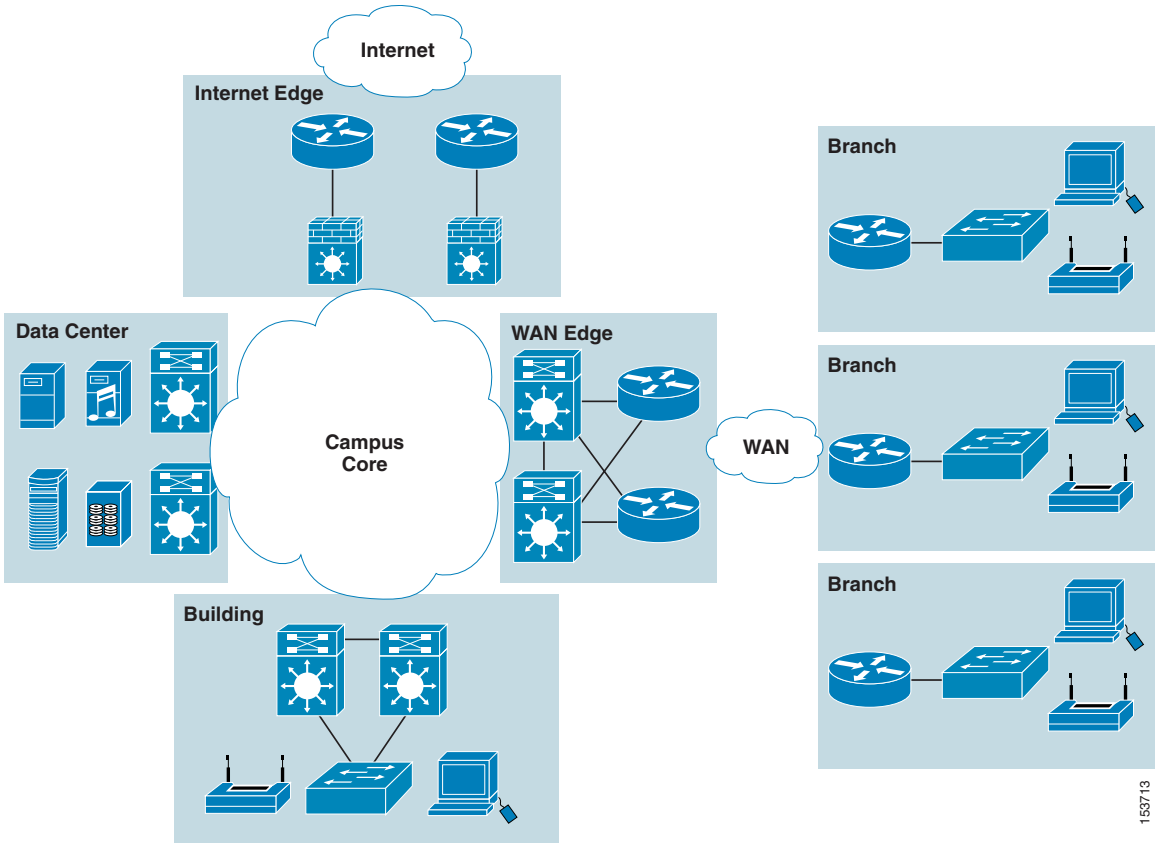
- ACLs do not support full data and control plane separation. Traffic originating from edge subnets that is associated to different user groups is sent to the core of the network and is handled in the common global routing table. This scenario is prone to configuration errors, which can cause the establishment of unwanted communications between different groups. Also, in cases where path differentiation must be achieved, using a common routing table forces the use of more complex configuration (such as the PBR described in Path Differentiation, page 17).
- In many cases, the configuration is simplified by assigning a dedicated (and possibly overlapping) IP address space to the subnets associated to different user groups. As previously described, this is usually not possible in a campus deployment because of route summarization requirements and because of the use of a shared global routing table.
- Depending on the IP addressing plan being used, the distributed ACL can become lengthy and require many statements to deny connectivity to the enterprise internal resources.

You can eliminate all the previously described limitations associated with using distributed ACLs if you can separate the data and control plans for each separate category of users. The following section describes a different network virtualization approach aimed at achieving this through the use of the Cisco VPN Routing and Forwarding (VRF) technology.

Path Isolation over the WAN using Distributed ACLs

The previous sections described the use of distributed ACLs to provide path isolation mechanisms to be implemented in a campus network to logically separate the traffic belonging to various categories of users. A similar scenario applies to the WAN when there is a need to extend the VPNs up to remote branch locations, as shown in Figure 9.

Figure 9 Connecting Branch Offices to the Main Campus



The various branch offices can connect to the WAN edge block of the campus network, either through a legacy WAN cloud (based, for example, on Frame Relay or ATM), or through an IP WAN cloud. In the second case, IPsec is more likely used to guarantee privacy of the traffic over the WAN. The details of IPsec deployments over the WAN are beyond the scope of this guide, but the following are some deployment alternatives:

- IPsec only
- IPsec with GRE
- IPsec with VTI
- DMVPN

Corresponding design guides can be found at the following URL:
http://www.cisco.com/warp/public/779/largeent/it/ese/srnd.html

The use of distributed ACLs to provide path isolation over the WAN presents the same characteristics and limitations described for the campus scenario in Path Isolation Using Distributed Access Control Lists, page 14. As a result, it is positioned again for applications requiring hub-and-spoke connectivity. The following assumptions are considered valid in this context:

- The hub resources are located in the main campus—These can be valid, for example, in the case of guest access if the access to the ISP is limited to the main campus and not available at the remote branch locations.
- The connectivity between the branch and the main campus is in place—This can either be unencrypted (legacy WAN based on Frame Relay or ATM) or encrypted. The details of this connectivity are beyond the scope of this guide.

In these scenarios, the format of the ACL that is required on the ISR router located at each branch location is identical to the one implemented in each campus distribution block, as follows:

```
ip access-list extended SEGM-RACL
 10 permit udp any any eq bootps
 20 permit udp any host <DNS-Server-IP> eq domain
 30 deny ip any <protected_prefixe>
 40 permit ip any <target_prefixes>
```

- Statements 10 and 20 allow connectivity to receive DHCP and DNS services (if needed).
- Statement 30 denies connectivity to protected resources that should not being accessed from this specific category of users.
- Statement 40 restricts connectivity only to the subset of required prefixes. The list of required prefixes can vary, depending on the specific application. For example, in the case of guest access, it can be all the public IP addresses representing the Internet, whereas for NAC remediation, it can be represented by the remediation server.

The RACL can be applied on all the router interfaces associated to each specific user group defined at the branch location. Only traffic directed to the specified target is allowed into the WAN toward the main campus.

Path Isolation using VRF-Lite and GRE

Connectivity Requirements

This particular solution is recommended in cases where there is a requirement for connectivity of many-to-one. This is most likely the scenario for applications such as guest access or NAC remediation, where the traffic originated on the edge of the network (campus buildings or branch offices) must be gathered to a centralized location (represented by the enterprise Internet edge or by the data center where a remediation server can be deployed).

In such scenarios, a hub-and-spoke topology is the recommended design. In a campus network, GRE tunnels can be used to transport the guest VLAN traffic from the first Layer 3 hop to a hub location, which is typically the Internet DMZ for an enterprise network. By placing the guest VLAN subnet (SVI) and the GRE interface into a VRF, you can separate the IP address space and routing from the rest of the enterprise network. Note that VRFs have to be defined only on the GRE tunnel endpoints (hub-and-spoke devices). One of the benefits of using GRE tunnels is that they can traverse multiple Layer 3 hops, but the VRF configuration is required only at the tunnel edges of the network.

A solution using GRE tunnels as a mechanism to segment the guest traffic has platform capability limitations. Table 2 provides a comparison of the GRE tunneling capabilities offered by the various Cisco switching platforms.

Table 2 GRE Support on Catalyst Switches

Platform	Supported	Implemented in Hardware
Catalyst 3560	No	N/A
Catalyst 3750	No	N/A
Catalyst 3750 Metro	No	N/A

Table 2 GRE Support on Catalyst Switches (continued)

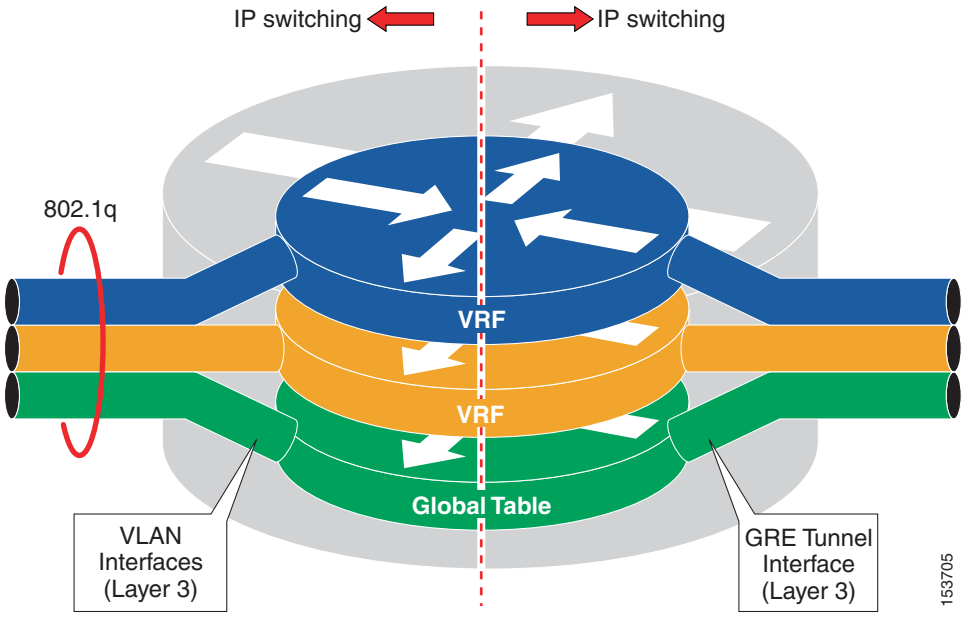
Platform	Supported	Implemented in Hardware
Catalyst 4500-SupII+/III/IV/V (4948)	Yes	No
Catalyst 6500-Sup2	Yes	No
Catalyst 6500-Sup720/Sup32	Yes	Yes

The information presented in Table 2 limits the applicability of this solution, depending on the specific Catalyst switches in place:

- In traditional designs, where the first Layer 3 hop is represented by the distribution layer devices, this approach is recommended when deploying a Catalyst 6500 with Sup720 or Sup32, because of the hardware-switching capability offered on these platforms. An exception to this recommendation can be for applications that do not require a large amount of bandwidth (such as guest access, where you might not want to provide large bandwidth). In that case, designs implementing the Catalyst 4500 in the distribution layer might be a candidate for this network virtualization solution. However, when originating (or terminating) GRE tunnels on a Catalyst 4500, it is a good practice to rate-limit the amount of GRE traffic that is allowed, to protect the CPU. More details on the configuration required for this are provided in QoS in Hub-and-Spoke Deployments, page 51.
- In routed access designs, where the demarcation line between Layer 2 and Layer 3 is moved down to the access layer, there are the following two scenarios:
  - The access layer contains deployed devices that support GRE (such as a Catalyst 6500 or 4500). In this case, GRE tunnels can be originated directly from the access layer devices, keeping in mind the bandwidth implications previously described when deploying platforms that do not support GRE in hardware.
  - The access layer contains deployed devices that do not support GRE (such as Catalyst 3xxx). In this scenario, GRE tunnels can be originated only from the distribution layer (assuming the platforms deployed there are GRE capable). As a result, some other mechanism should be deployed to maintain the logical separation of traffic for different user groups between the access and distribution layers. One possible way to achieve this is to use VRF-Lite with dot1q trunking.

Figure 10 shows the definition of various VRFs on the distribution layer device, with the corresponding mapping to the VRF for the VLANs defined on the Layer 2 domain of the network and the GRE tunnels part of the Layer 3 domain.

Figure 10 VRF-Lite



The diagram in Figure 10 is valid for both traditional and routed access designs when GRE tunnels are originated on the distribution layer switches. When deploying routed access designs where GRE tunnels can be originated from the access layer devices, the only difference is the absence of the trunk connection on the left, because each switch port is mapped to a specific VLAN.

To deploy end-to-end network virtualization across the network, a mapping between VLANs to VRFs and then VRFs to GRE on one side, as well as between the GRE tunnel interfaces and VRFs on the other side is required. The next two paragraphs provide a more detailed description of the configuration required to implement this form of traffic isolation.

Configuration Details

This section describes two options to build logical overlay networks using GRE and VRF. The first approach uses point-to-point GRE connections between devices, and the second one introduces the use of mGRE interfaces. The use of mGRE technology is particularly suited for applications requiring hub-and-spoke connectivity, as described in this section.

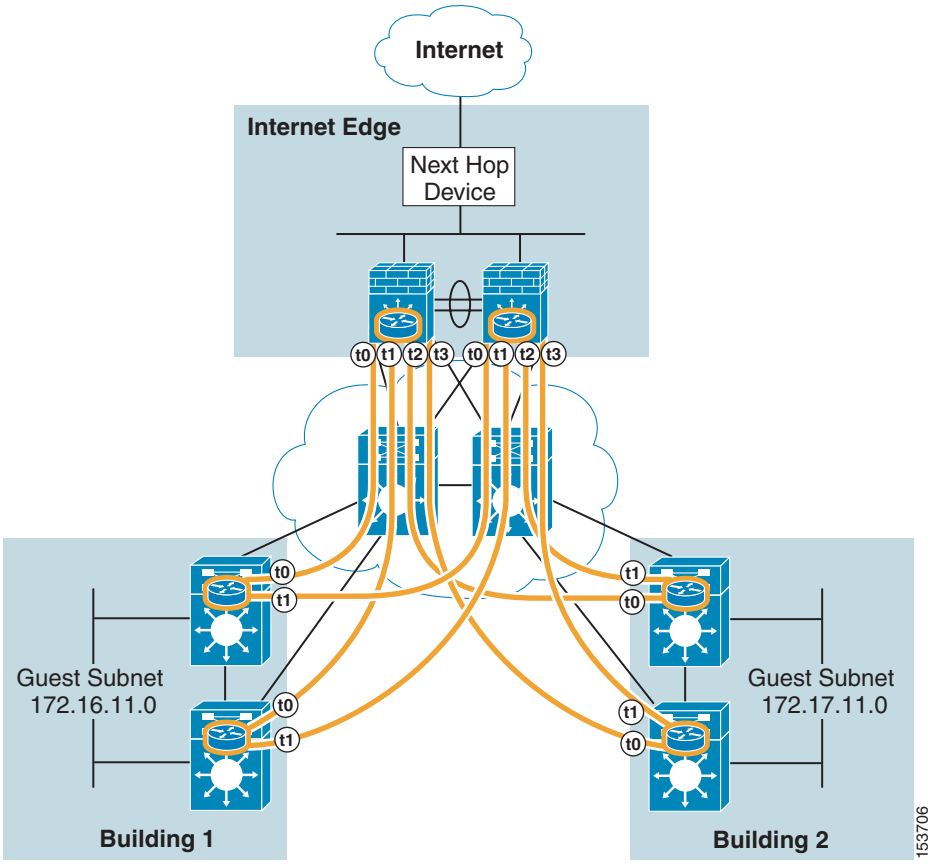
Using Point-to-Point GRE

The traditional configuration for GRE tunnels requires the creation of point-to-point tunnel interfaces on both sides of the tunnel. When building a hub-and-spoke topology, the use of point-to-point GRE tunnels requires that you to create a separate logical interface on the hub switches every time a new spoke needs to be added. This is both configuration-intensive and router resource-intensive. To address the performance considerations, Cisco recommends using a Catalyst 6500 with a Supervisor 720 that has GRE support in hardware. To address the configuration challenges associated with supporting multiple GRE tunnels at the hub site, an alternative network design based on mGRE and Next Hop Resolution Protocol (NHRP) is introduced. However, in some cases, point-to-point GRE might be the only option because mGRE and NHRP are not supported on all platforms (for example, they are not supported on Catalyst 4500 switches).

The following configuration steps accompany the network diagram shown in Figure 11. Keep in mind the following considerations when considering the required configuration:

- The example is valid for a guest access application, so point-to-point GRE tunnels are defined between a generic spoke device and the centralized hub in the Internet edge. Also, traffic is originated from guest subnets defined at the edge of the network (spokes).
- The configuration sample refers to the traditional campus design, so VRF and GRE are defined on the distribution layer devices.
- Catalyst 6500 switches are deployed as spoke and hub devices. The Catalyst 4500 is also a viable alternative for applications not requiring high throughput.
- It is assumed that all traffic directed to the Internet is sent to an undefined next hop device. Depending on the specific application, this device might be an appliance, such as a firewall or a router.

Figure 11 Hub-and-Spoke with Point-to-Point GRE Tunnels



Note The following configuration sections assume that basic network connectivity (for example, in the global routing table) is already in place in the network.

Hub GRE Configuration

On each hub device, a separate tunnel (and corresponding loopback) interface is required for each spoke switch. In the previous example, there are four spokes devices, representing the two pairs of distribution layer switches for two campus buildings.

Note The configuration samples in the following sections refer specifically to a guest access deployment. However, they are also valid for all applications requiring hub-and-spoke connectivity.

```
ip vrf guest
 rd 100:1
!
interface Loopback0
 description src GRE p2p tunnel 1
 ip address 10.122.200.1 255.255.255.255
!
interface Loopback1
 description src GRE p2p tunnel 2
 ip address 10.122.200.2 255.255.255.255
!
interface Loopback2
 description src GRE p2p tunnel 3
 ip address 10.122.200.3 255.255.255.255
!
interface Loopback3
 description src GRE p2p tunnel 4
 ip address 10.122.200.4 255.255.255.255
!
interface Tunnel0
 description GRE p2p tunnel 1
 ip vrf forwarding guest
 ip address 172.32.1.1 255.255.255.252
 tunnel source Loopback0
 tunnel destination 10.122.210.1
!
interface Tunnel1
 description GRE p2p tunnel 2
 ip vrf forwarding guest
 ip address 172.32.1.5 255.255.255.252
 tunnel source Loopback1
 tunnel destination 10.122.210.2
!
interface Tunnel2
 description GRE p2p tunnel 3
 ip vrf forwarding guest
 ip address 172.32.1.9 255.255.255.252
 tunnel source Loopback2
 tunnel destination 10.122.210.3
!
interface Tunnel3
 description GRE p2p tunnel 4
 ip vrf forwarding guest
 ip address 172.32.1.13 255.255.255.252
 tunnel source Loopback3
 tunnel destination 10.122.210.4
```

Note that each tunnel interface is mapped to the guest VRF using the **ip vrf forwarding** command, which is the key starting point in building the overlay logical network. The use of VRF allows great flexibility when planning the IP addressing for the guest subnets. In the preceding example, the overlay logical network is using a 172.16.0.0 address space, whereas all the addresses used in the global table (loopback

interfaces, and so on) are part of the 10.0.0.0/8 subnet. This means that the IP addresses assigned to each defined user group can be independently selected from the block of addresses associated to that specific building block in the global table. Overlapping IP address space is also supported on different VRFs. For example, network 10.1.1.0/24 can exist in multiple VRFs in multiple locations.

The addresses to be used for the loopback interfaces used as source and destination of the GRE traffic should be carefully selected to avoid the creation of routing black holes. See Loopback IP Address Considerations, page 39 for more information on this subject.

Spoke GRE Configuration

The configuration required on each spoke is very similar to the one described previously: two tunnel interfaces are configured to connect to the pair of redundant hub devices in the Internet edge block. Referring to Figure 11, the following configuration sample is valid for one of the two spokes in Building 1:

```
ip vrf guest
 rd 100:1
!
interface Loopback0
 description src GRE tunnel to hub-1
 ip address 10.122.210.1 255.255.255.255
!
interface Loopback1
 description src GRE tunnel to hub-2
 ip address 10.122.211.1 255.255.255.255
!
interface Tunnel0
 description GRE tunnel to hub-1
 ip vrf forwarding guest
 ip address 172.32.1.2 255.255.255.252
 tunnel source Loopback0
 tunnel destination 10.122.200.1
!
interface Tunnel1
 description GRE tunnel to hub-2
 ip vrf forwarding guest
 ip address 172.32.2.2 255.255.255.252
 tunnel source Loopback1
 tunnel destination 10.122.201.1
```

Again, the logical tunnel interfaces must be mapped to the VRF to force the Internet-bound guest traffic into the GRE tunnel that carries the traffic to the Internet edge at the hub site. All the guest traffic originates from users deployed in a dedicated guest VLAN (at least for wired users, as previously described). To maintain an end-to-end segregation of guest traffic, the corresponding VLAN interface (logical SVI) must also be mapped to the guest VRF, as shown in the following configuration sample.

Note Typical deployments have more than one guest VLAN defined for each campus distribution block. In this case, all the corresponding VLAN interfaces must be mapped to the same VRF.

```
interface Vlan11
 description Wired Guest subnet
 ip vrf forwarding guest
 ip address 172.16.11.2 255.255.255.0
 ip helper-address 172.18.2.10
 standby 11 ip 172.16.11.1
 standby 11 timers msec 250 msec 800
 standby 11 priority 105
 standby 11 preempt delay minimum 180
```

Note HSRP (or any other redundant gateway protocol) is relevant when deploying traditional campus designs, where the demarcation line between Layer 2 and Layer 3 is placed in the distribution layer switches. HSRP is not needed in a routed access scenario.

Enabling a Routing Protocol

When the VRFs identifying the same user group have been linked together by GRE tunnels (creating the logical overlay network), it is time to start entering routing information into the routing tables for each defined group. The easiest way to do this is through static routing; a default static route pointing to the hubs can be configured on each spoke device. In this way, all the traffic originating, for example, from the guest subnets and directed to the Internet is GRE-encapsulated and conveyed toward the enterprise Internet edge.

The use of static routing also requires the configuration of specific static routes on the hub to allow return traffic directed to the edge subnets. Introducing a dynamic routing protocol in the overlay network brings the following two main advantages:

- The routing updates serve as keepalives for the GRE tunnels. The devices use the GRE interfaces to send traffic only if valid routing information is received, which ensures network connectivity across the tunnel.
- When supporting redundant GRE uplinks, load balancing of traffic and resiliency are automatically achieved by using the routing protocol characteristics.

The configuration details for how to enable a routing protocol in the context of a specific VRF differ, depending on the chosen protocol. Some routing protocols (such as EIGRP and BGP) introduce the concept of address families. The idea is to have a single routing process running on the device and to define a separate address family that is mapped to each VRF. Other routing protocols (such as OSPF) allow a different routing process for each VRF to be created.

This guide considers EIGRP and OSPF because they are the most common routing protocols found in enterprise networks. The following configuration samples refer to the same network diagram shown in Figure 11.

Note The routing protocol enabled in the context of each VRF is totally independent from the IGP running in the other VRFs or in the global routing table.

EIGRP

To run EIGRP in the context of a VRF, the VRF-specific address family needs to be configured. The configuration is slightly different for hub-and-spoke because the hubs must also advertise a default route to the spokes. Because of this default route, all the traffic that originates from the edge subnets is forced to the hubs.

The static default route configured on the hub is pointing to the next hop device shown in Figure 11.

- Hub

```
ip route vrf guest 0.0.0.0 0.0.0.0 172.18.1.30
!
router eigrp 100
 passive-interface default
 no passive-interface Tunnel0
 no passive-interface Tunnel1
 no passive-interface Tunnel2
 no passive-interface Tunnel3
 no auto-summary
```

```
 !
 address-family ipv4 vrf guest
 redistribute static metric 1000000 500 255 1 1500
 network 172.32.1.0 0.0.0.255
 no auto-summary
 autonomous-system 100
 exit-address-family
Spoke
router eigrp 100
 passive-interface default
 no passive-interface Tunnel0
 no passive-interface Tunnel1
 no auto-summary
  !
 address-family ipv4 vrf guest
network 172.16.100.0 0.0.0.255
network 172.16.200.0 0.0.0.255
  network 172.16.11.0 0.0.0.255
  no auto-summary
  autonomous-system 100
  exit-address-family
```

This design is resilient because each spoke receives a redundant default route to each of the hubs located in the Internet edge. Each hub learns the guest subnets from each spoke. Note how, by default, the spoke learns not only the default routes from each hub, but also the subnet information of other guests.

- Spoke

```
6500-1-Bldg#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.32.1.1 to network 0.0.0.0
     172.17.0.0/24 is subnetted, 1 subnets
D       172.17.11.0 [90/310044672] via 172.32.1.1, 00:00:35, Tunnel0
                    [90/310044672] via 172.32.2.1, 00:00:35, Tunnel1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.11.0 is directly connected, Vlan11
     172.32.0.0/30 is subnetted, 8 subnets
D       172.32.1.12 [90/310044416] via 172.32.1.1, 00:00:47, Tunnel0
D       172.32.2.12 [90/310044416] via 172.32.2.1, 00:00:35, Tunnel1
D       172.32.1.8 [90/310044416] via 172.32.1.1, 00:00:53, Tunnel0
D       172.32.2.8 [90/310044416] via 172.32.2.1, 00:00:44, Tunnel1
D       172.32.1.4 [90/310044416] via 172.32.1.1, 00:00:48, Tunnel0
D       172.32.2.4 [90/310044416] via 172.32.2.1, 00:00:41, Tunnel1
C       172.32.1.0 is directly connected, Tunnel0
C       172.32.2.0 is directly connected, Tunnel1
D*EX 0.0.0.0/0 [170/297372416] via 172.32.1.1, 00:00:55, Tunnel0
               [170/297372416] via 172.32.2.1, 00:00:55, Tunnel1
```

To get to a situation where the spokes have only the default routes in their routing tables, some additional configuration is required. For example, it is possible to apply an outbound filter on the hub to advertise only the default route toward each spoke. This is achieved by the following configuration:

```
ip access-list standard default-only
 permit 0.0.0.0
!
router eigrp 100
```

```
 address-family ipv4 vrf guest
 distribute-list default-only out
```

The result of this configuration on the spoke routing table is as follows:

```
6500-1-Bldg#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.32.2.1 to network 0.0.0.0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.11.0 is directly connected, Vlan11
     172.32.0.0/30 is subnetted, 2 subnets
C       172.32.1.0 is directly connected, Tunnel0
C       172.32.2.0 is directly connected, Tunnel1
D*EX 0.0.0.0/0 [170/297372416] via 172.32.2.1, 00:00:32, Tunnel1
               [170/297372416] via 172.32.1.1, 00:00:32, Tunnel0
```

Differently from the spokes, to be able to properly route return traffic, the two hubs must contain information about all the guest subnets that are deployed in the campus in their routing tables. Referring to the example in Figure 11, the routing table on each hub device appears like the following example.

```
6500-Int-1#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.18.1.30 to network 0.0.0.0
     172.17.0.0/24 is subnetted, 1 subnets
D       172.17.11.0 [90/15360256] via 172.32.10.4, 00:01:10, Tunnel2
                    [90/15360256] via 172.32.10.5, 00:01:10, Tunnel3
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.11.0 [90/15360256] via 172.32.10.2, 00:00:10, Tunnel0
                    [90/15360256] via 172.32.10.3, 00:00:10, Tunnel1     172.18.0.0/24 is
subnetted, 1 subnets
C       172.18.1.0 is directly connected, Vlan181
     172.32.0.0/30 is subnetted, 8 subnets
C       172.32.1.12 is directly connected, Tunnel3
D       172.32.2.12 [90/310044416] via 172.32.1.14, 00:02:15, Tunnel3
C       172.32.1.8 is directly connected, Tunnel2
D       172.32.2.8 [90/310044416] via 172.32.1.10, 00:02:30, Tunnel2
C       172.32.1.4 is directly connected, Tunnel1
D       172.32.2.4 [90/310044416] via 172.32.1.6, 00:02:44, Tunnel1
C       172.32.1.0 is directly connected, Tunnel0
D       172.32.2.0 [90/310044416] via 172.32.1.2, 00:02:56, Tunnel0
S*   0.0.0.0/0 [1/0] via 172.18.1.30
```

As shown, each hub has a redundant path to the route aggregate advertised from each building block. As a result, even if each spoke has no knowledge of the other guest subnets defined across the campus network, communication between them is still possible because the hub has the information to route these packets in its routing table. The advantage in building this hub-and-spoke overlay network is that

policy enforcement to deny communications between guest subnets defined in separate campus buildings can be centralized on the two hub devices, and it is not required to be distributed on each spoke at the edge of the network.

To limit communication between guest subnets defined in the same campus building, the policy needs to be applied on the first L3 hop device, represented by the distribution layer switch (for traditional L2/L3 campus designs) or by the access layer switch (for the routed access campus design).

OSPF

Differently from EIGRP, there is no concept of an address family in OSPF. To enable OSPF in the context of a VRF, you must define a new process and bind it to the specific VRF:

- Hub

```
ip route vrf guest 0.0.0.0 0.0.0.0 172.18.1.30
!
router ospf 1 vrf guest
 log-adjacency-changes
 passive-interface default
 no passive-interface Tunnel0
 no passive-interface Tunnel1
 no passive-interface Tunnel2
 no passive-interface Tunnel3
 network 172.32.1.0 0.0.0.255 area 0
 default-information originate
```

- Spoke

```
router ospf 1 vrf guest
 log-adjacency-changes
 passive-interface default
 no passive-interface Tunnel0
 no passive-interface Tunnel1
 network 172.16.11.0 0.0.0.255 area 16
 network 172.32.1.0 0.0.0.255 area 0
 network 172.32.2.0 0.0.0.255 area 0
```

As described for EIGRP, the configuration causes the spoke routers to have information about all the guest subnets in their routing table, as in the following example:

```
6500-1-Bldg#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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

Gateway of last resort is 172.32.2.1 to network 0.0.0.0

     172.17.0.0/24 is subnetted, 1 subnets
O       172.17.11.0 [110/22223] via 172.32.1.1, 00:00:05, Tunnel0
                    [110/22223] via 172.32.2.1, 00:00:05, Tunnel1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.11.0 is directly connected, Vlan11
     172.32.0.0/30 is subnetted, 8 subnets
O       172.32.1.12 [110/22222] via 172.32.1.1, 00:00:05, Tunnel0
O       172.32.2.12 [110/22222] via 172.32.2.1, 00:00:05, Tunnel1
O       172.32.1.8 [110/22222] via 172.32.1.1, 00:00:06, Tunnel0
O       172.32.2.8 [110/22222] via 172.32.2.1, 00:00:06, Tunnel1
O       172.32.1.4 [110/22222] via 172.32.1.1, 00:00:06, Tunnel0
```

```
O       172.32.2.4 [110/22222] via 172.32.2.1, 00:00:07, Tunnel1
C       172.32.1.0 is directly connected, Tunnel0
C       172.32.2.0 is directly connected, Tunnel1
O*E2 0.0.0.0/0 [110/1] via 172.32.2.1, 00:00:07, Tunnel1
               [110/1] via 172.32.1.1, 00:00:07, Tunnel0
```

Similarly to the EIGRP example, it is possible to apply a distribute list statement to eliminate these routes from the spoke devices and to import only a default route. In the OSPF scenario, this filter should be configured on each spoke (and not on the hub) because each router configured for OSPF must maintain a common topology database. This is achieved with the following configuration:

```
ip access-list standard default-only
 permit 0.0.0.0
!
router ospf 1 vrf guest
distribute-list default-only in
```

As a result of this configuration, the spoke eventually learns (in the routing table) only a default route pointing to the Internet edge, as follows:

```
6500-1-Bldg#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.32.2.1 to network 0.0.0.0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.11.0 is directly connected, Vlan11
     172.32.0.0/30 is subnetted, 2 subnets
C       172.32.1.0 is directly connected, Tunnel0
C       172.32.2.0 is directly connected, Tunnel1
O*E2 0.0.0.0/0 [110/1] via 172.32.2.1, 00:00:23, Tunnel1
               [110/1] via 172.32.1.1, 00:00:23, Tunnel0
```

From the point of view of the hub, the routing table appears similar to the EIGRP scenario. The hub has knowledge of all the guest subnets defined around the campus, so some centralized policy configuration might be required to prevent inter-guest communications.

```
6500-Int-1#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.18.1.30 to network 0.0.0.0
     172.17.0.0/24 is subnetted, 1 subnets
O IA 172.17.11.0/24 [110/11112] via 172.32.1.10, 00:03:08, Tunnel2
                    [110/11112] via 172.32.1.14, 00:03:08, Tunnel3
     172.16.0.0/24 is subnetted, 1 subnets
O IA 172.16.11.0/24 [110/11112] via 172.32.1.2, 00:03:08, Tunnel0
                    [110/11112] via 172.32.1.6, 00:03:08, Tunnel1
     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.1.0 is directly connected, Vlan181
     172.32.0.0/30 is subnetted, 8 subnets
C       172.32.1.12 is directly connected, Tunnel3
O       172.32.2.12 [110/22222] via 172.32.1.14, 00:03:08, Tunnel3
```

```
C       172.32.1.8 is directly connected, Tunnel2
O       172.32.2.8 [110/22222] via 172.32.1.10, 00:03:08, Tunnel2
C       172.32.1.4 is directly connected, Tunnel1
O       172.32.2.4 [110/22222] via 172.32.1.6, 00:03:12, Tunnel1
C       172.32.1.0 is directly connected, Tunnel0
O       172.32.2.0 [110/22222] via 172.32.1.2, 00:03:12, Tunnel0
S*   0.0.0.0/0 [1/0] via 172.18.1.30
```

Note Cisco does not recommend applying a distribute list statement on the spoke devices, because doing so causes a discrepancy between the content of the topology database and the routing table.

Using mGRE Technology

When compared to the point-to-point GRE scenario described in the previous section, the use of mGRE interfaces on the hub switches has several advantages:

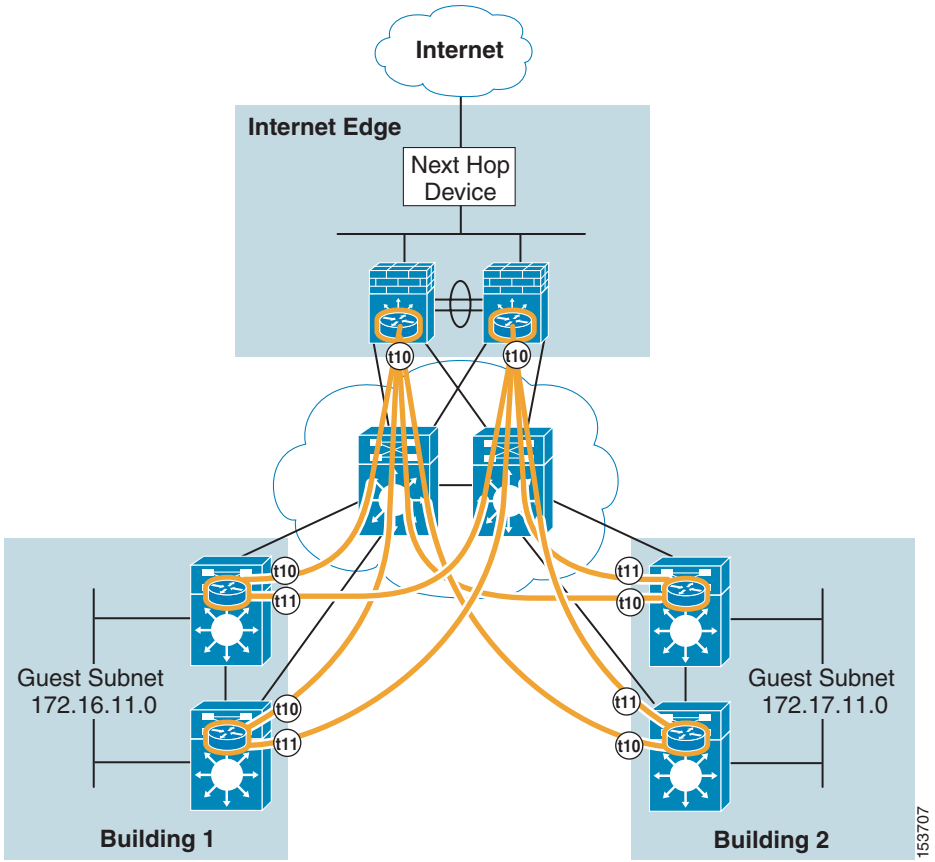
- Simplified configuration on the hub—Only one loopback and one tunnel interface are required, instead of configuring a pair for each spoke device. To connect to multiple edge devices, the tunnel interface works in mGRE mode.
- Dynamic addition of spoke devices—New spokes can be added without requiring any configuration changes on the hub devices.
- Simplified IP addressing—The overlay logical mGRE network is part of a single IP subnet and many distinct point-to-point subnets are not required for each GRE spoke tunnel.

At the same time, an additional mechanism is needed, NHRP, to allow the hub devices to dynamically discover the spokes and establish GRE tunnels with them. NHRP, as defined in RFC 2332, is a Layer 2 address resolution protocol and cache, similar to Address Resolution Protocol (ARP) and Frame Relay inverse-ARP. When a tunnel interface is an mGRE, NHRP tells the mGRE process where to tunnel a packet to reach a certain address. NHRP is a client-server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database where it registers each spoke, the mapping between the physical address (used as GRE tunnel destination), and the logical address assigned to the spoke tunnel interface. Each spoke provides this information to the hub, sending an NHRP registration message at startup time.

Note Support for NHRP in the context of a VRF is restricted to Catalyst 6500 platforms with Sup720 and Sup32 running software release 12.2(18)SXE and later. This implies that to deploy the solution described in this section, these devices must both be deployed at the hub-and-spoke locations.

Following are the configuration steps required for creating the hub-and-spoke overlay network using mGRE interfaces on the hub devices (see Figure 12).

Figure 12 Hub-and-Spoke Using mGRE Technology



Similarly to the point-to-point scenario, the following considerations are valid in this instance:

- The example is valid for a guest access application, so point-to-point GRE interfaces are defined for each spoke device, whereas mGRE is used on the centralized hub in the Internet edge. Also, traffic is originated from guest subnets defined at the edge of the network (spokes).
- The configuration sample refers to the traditional campus design, so VRF and GRE are defined on the distribution layer devices.
- Catalyst 6500 switches are deployed both as spoke and hub devices. Catalyst 4500s are not a viable alternative for this design because of the lack of support of NHRP in the context of the VRF.

Hub mGRE Configuration

The configuration required to create an mGRE interface on the hub and enable the NHRP functionality is as follows:

```
ip vrf guest
 rd 100:1
!
interface Loopback10
 description src mGRE tunnel for Guest
 ip address 10.122.200.10 255.255.255.255
!
interface Tunnel10
 description mGRE tunnel for Guest
 ip vrf forwarding guest
 ip address 172.32.10.1 255.255.255.0
```

```
 no ip redirects
 ip nhrp map multicast dynamic
 ip nhrp network-id 10
 tunnel source Loopback10
 tunnel mode gre multipoint
```

NHRP is enabled on the mGRE interface using the **ip nhrp network-id** command. The value specified must match the one configured on the spoke devices. Also, the **ip nhrp map multicast dynamic** command is required to enable dynamic routing protocols to work over the mGRE tunnel when IGP routing protocols use multicast packets. The **dynamic** keyword prevents the hub device from needing a separate configuration line for a multicast mapping for each spoke router. This is important because the goal is to avoid any reconfiguration of the hub devices when adding a new spoke component.

Spoke GRE Configuration

The configuration of the spoke devices is almost identical to the one previously described for the point-to-point scenario. The only difference is the addition of the NHRP-related commands:

```
ip vrf guest
 rd 100:1
!
interface Loopback10
 description src GRE tunnel for Guest to hub-1
 ip address 10.122.210.10 255.255.255.255
!
interface Loopback11
 description src GRE tunnel for Guest to hub-2
 ip address 10.122.211.10 255.255.255.255
!
interface Tunnel10
 description GRE tunnel for Guest to hub-1
 ip vrf forwarding guest
 ip address 172.32.10.2 255.255.255.0
 ip nhrp map 172.32.10.1 10.122.200.10
 ip nhrp network-id 10
 ip nhrp nhs 172.32.10.1
 ip nhrp registration timeout 60
 tunnel source Loopback10
 tunnel destination 10.122.200.10
!
interface Tunnel11
 description GRE tunnel for Guest to hub-2
 ip vrf forwarding guest
 ip address 172.32.11.2 255.255.255.0
 ip nhrp map 172.32.11.1 10.122.201.10
 ip nhrp network-id 11
 ip nhrp nhs 172.32.11.1
 ip nhrp registration timeout 60
 tunnel source Loopback11
 tunnel destination 10.122.201.10
```

Similarly to the hub case, the **ip nhrp network-id** command is used to enable the NHRP process on the tunnel interfaces (the values specified must match the values configured on the two hubs). In addition to that, the **ip nhrp nhs** command is required to specify the address of the NHRP server (hub). Finally, the **ip nhrp registration timeout** command is required to tune the frequency (in seconds) at which the spokes send the NHRP registration messages to the hubs. This command is required to allow a spoke to re-register in case the connectivity with the hub is interrupted and restored, which occurs every 2400 seconds by default.

Note The **ip nhrp map multicast** command is not required on the spoke devices because the tunnel interface is point-to-point, so all multicast packets are automatically sent to the other end (hub).

As described in Using Point-to-Point GRE , page 23, a mapping from the logical VLAN interface defining the guest subnets and the guest VRF is also required:

```
interface Vlan11
 description Wired Guest subnet
 ip vrf forwarding guest
 ip address 172.16.11.2 255.255.255.0
 standby 11 ip 172.16.11.1
 standby 11 timers msec 250 msec 800
 standby 11 priority 105
 standby 11 preempt delay minimum 180
```

Verifying the NHRP Information

After configuring the tunnel interfaces on the hub-and-spoke, it should be possible to verify that the hub is receiving the NHRP registration message from the spoke device, therefore adding dynamic entries to the NHRP cache:

```
6500-Int-1#sh ip nhrp
172.32.10.2/32 via 172.32.10.2, Tunnel10 created 00:01:52, expire 01:59:05
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 10.122.210.10
172.32.10.3/32 via 172.32.10.3, Tunnel10 created 00:01:03, expire 01:59:54
  Type: dynamic, Flags: authoritative unique registered used
  NBMA address: 10.122.210.11
172.32.10.4/32 via 172.32.10.4, Tunnel10 created 00:00:33, expire 01:59:26
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 10.122.210.12
172.32.10.5/32 via 172.32.10.5, Tunnel10 created 00:00:06, expire 01:59:56
  Type: dynamic, Flags: authoritative unique registered used
  NBMA address: 10.122.210.13
```

As shown in the previous configuration sample, the hub learns the physical, non-broadcast multiaccess address (NBMA) used to tunnel GRE packets destined to the spoke. This information is refreshed by the spoke with NHRP registration messages every 60 seconds (because of the tuning done with the **ip nhrp registration timeout** command). The default expiration time (hold time) is 7200 seconds (two hours) as noted on the right side in this example (expire 01:59:05). Under normal circumstances, this value should never go below 01:59:00, because it is re-initialized by the receipt of NHRP registration messages every 60 seconds.

Enabling a Routing Protocol

From a topology perspective, the routing protocol runs only between the spoke router and one or more hub devices. The solution implementing mGRE interfaces has been tested with EIGRP and OSPF because they are the most commonly deployed choices among enterprise customers.

When the connection of the spoke to the network comes up, it is ready to begin transmitting routing protocol information because the tunnel interface is configured as point-to-point. On the other side, the hub device cannot begin sending routing protocol information until NHRP registrations arrive from each spoke device and the NHRP cache gets populated.

Consider the following when configuring routing protocols in this scenario:

- GRE tunnel bandwidth—The default bandwidth of a GRE tunnel is 9 Kbps, which has two unwanted consequences:

  - Any routing protocol using bandwidth as a metric is being given misleading information, which can cause unpredictable results.
  - Cisco EIGRP assigns half of this bandwidth for the use of the routing protocol, which most likely is insufficient.

  Cisco recommends configuring the bandwidth parameter on GRE tunnel interfaces to the actual bandwidth available on the link.

- IP maximum transmission unit (MTU)—It is important, especially when using OSPF, to verify that the IP MTU settings match on the tunnel interfaces on both sides of the link. The MTU value recommended here is 1400 bytes, which leaves room for GRE and IPsec overhead (if needed) and avoids packet fragmentation. More information on this topic can be found in Verifying the NHRP Information, page 35.
- OSPF interface types and priority—In the hub-and-spoke topology previously described, the mGRE tunnel interface is considered point-to-point from an OSPF standpoint. Because the same interface starts receiving hellos and OSPF packets from different spokes, this prevents the establishment of adjacencies. To fix the problem, configure the OSPF network type as broadcast on both the hubs and all the spokes. Also, set the OSPF priority to 0 on the spokes to guarantee that the hubs become the designated router (DR) and the backup designated router (BDR).

Based on these considerations, the configuration of the generic hub-and-spoke GRE interfaces needs to be changed as follows. (This configuration sample is also valid for OSPF.)

- Hub

```
interface Tunnel10
 description mGRE tunnel
 bandwidth 1000
 ip vrf forwarding guest
 ip address 172.32.10.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip ospf network broadcast
 tunnel source Loopback0
 tunnel mode gre multipoint
```

- Spoke

```
interface Tunnel10
 description GRE tunnel for Guest to hub-1
 bandwidth 1000
 ip vrf forwarding guest
 ip address 172.32.10.2 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 10
 ip nhrp nhs 172.32.10.1
 ip nhrp registration timeout 60
 ip ospf network broadcast
 ip ospf priority 0
 tunnel source Loopback10
 tunnel destination 10.122.200.10
!
interface Tunnel11
 description GRE tunnel for Guest to hub-2
 bandwidth 1000
 ip vrf forwarding guest
 ip address 172.32.11.2 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 11
```

```
ip nhrp nhs 172.32.11.1
ip nhrp registration timeout 60
ip ospf network broadcast
ip ospf priority 0
tunnel source Loopback11
tunnel destination 10.122.201.10
```

The configuration required to enable the routing protocols in the context of the guest VRF is identical to that described in the point-to-point scenario.

The only difference in this case is the fact that the hub devices learn all the routes for the guest subnets out of the same mGRE interface. Because of the additional information contained in the NHRP cache, the hubs are able to route back the traffic to the proper spokes (see the following sample output for an EIGRP example).

```
6500-Int-1#sh ip route vrf guest
Routing Table: guest
Codes: 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, E - EGP
       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
Gateway of last resort is 172.18.1.30 to network 0.0.0.0
     172.17.0.0/24 is subnetted, 1 subnets
D       172.17.11.0 [90/15360256] via 172.32.10.4, 00:01:10, Tunnel10
                    [90/15360256] via 172.32.10.5, 00:01:10, Tunnel10
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.11.0 [90/15360256] via 172.32.10.2, 00:00:10, Tunnel10
                    [90/15360256] via 172.32.10.3, 00:00:10, Tunnel10     172.18.0.0/24 is
subnetted, 1 subnets
C       172.18.1.0 is directly connected, Vlan181
     172.32.0.0/16 is variably subnetted, 10 subnets, 2 masks
C       172.32.1.12/30 is directly connected, Tunnel3
D       172.32.2.12/30 [90/310044416] via 172.32.10.5, 00:00:38, Tunnel10
C       172.32.1.8/30 is directly connected, Tunnel2
C       172.32.10.0/24 is directly connected, Tunnel10
D       172.32.2.8/30 [90/310044416] via 172.32.10.4, 00:00:39, Tunnel10
D       172.32.11.0/24 [90/28160000] via 172.32.10.2, 00:00:35, Tunnel10
                       [90/28160000] via 172.32.10.5, 00:00:35, Tunnel10
                       [90/28160000] via 172.32.10.4, 00:00:35, Tunnel10
                       [90/28160000] via 172.32.10.3, 00:00:35, Tunnel10
C       172.32.1.4/30 is directly connected, Tunnel1
D       172.32.2.4/30 [90/310044416] via 172.32.10.3, 00:00:35, Tunnel10
C       172.32.1.0/30 is directly connected, Tunnel0
D       172.32.2.0/30 [90/310044416] via 172.32.10.2, 00:00:40, Tunnel10
S*   0.0.0.0/0 [1/0] via 172.18.1.30
```

MTU Considerations

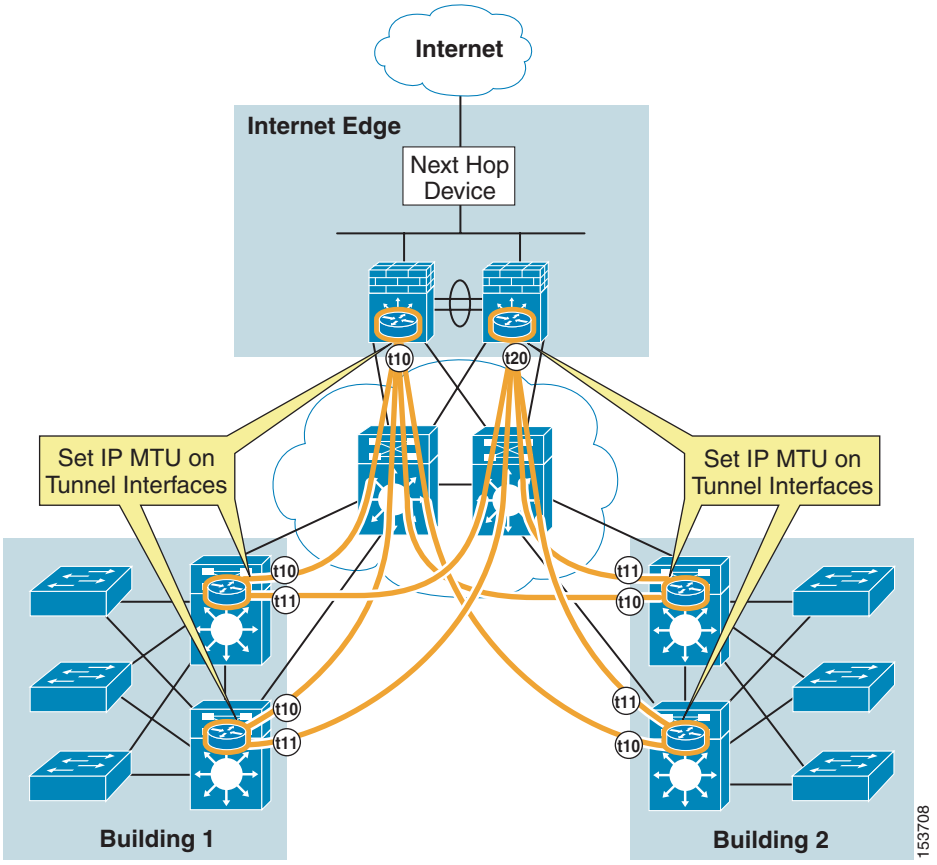
The use of GRE tunnels to create overlay logical networks can eventually cause MTU issues because of the increased size of the IP packets. The goal is to avoid IP fragmentation whenever possible, and to avoid all related issues. For more information, see the following URL: http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper09186a00800d6979.shtml.

Fragmentation at the endpoints of a TCP connection is avoided by the negotiation of TCP maximum segment size (MSS) performed by the same endpoint stations. However, TCP MSS cannot help in avoiding fragmentation happening in the path between the endpoints. This can be because of the existence of a smaller MTU link or, as it is in this case, because of the need to tunnel IP packets that can render their size larger than the original.

To deal with this problem, two approaches are described in this guide. The first is based on the use of Path MTU Discovery (PMTUD), which allows you to dynamically determine the lowest MTU along the path from a packet source to its destination. Hosts usually perform PMTUD by default by the Do Not Fragment (DF) bit being set in all the sourced TCP/IP packets. With the DF bit set, if a router along the path tries to forward an IP datagram to a link that has a lower MTU than the size of the packet, the router drops the packet and returns an Internet Control Message Protocol (ICMP) Destination Unreachable message to the source of this IP datagram, with the code indicating fragmentation needed and DF set (type 3, code 4). When the source station receives the ICMP message, it lowers the send message segment size (MSS), and when TCP retransmits the segment, it uses the smaller segment size. This process continues until the correct MSS to allow end-to-end communication is determined.

When deploying hub-and-spoke overlay networks in a campus environment, the recommended approach is to configure a lower MTU value on the GRE tunnel interfaces for both the hub and spoke devices, as shown in Figure 13.

Figure 13 Setting IP MTU on GRE Interfaces



The corresponding required configuration is as follows:

```
interface Tunnel11
 description GRE tunnel for Guest to hub-2
```

```
ip mtu 1400
```

The value 1400 is small enough to accommodate the GRE header and, eventually, also for an additional IPsec header that might be used when encrypting the traffic.

There may be scenarios where PMTUD fails to accomplish what is described in the previous section. The most common reason for this is when the ICMP messages are blocked by a router or firewall that is positioned between this router and the sender. However, this should not be the case when deploying overlay networks to enable communication in a campus environment. For some workarounds in cases where the failure of PMTUD prevents communications between endpoints connected to the campus network and the Internet, see the following URL: http://www.cisco.com/en/US/partner/tech/tk827/tk369/technologies_tech_note09186a0080093f1f.shtml

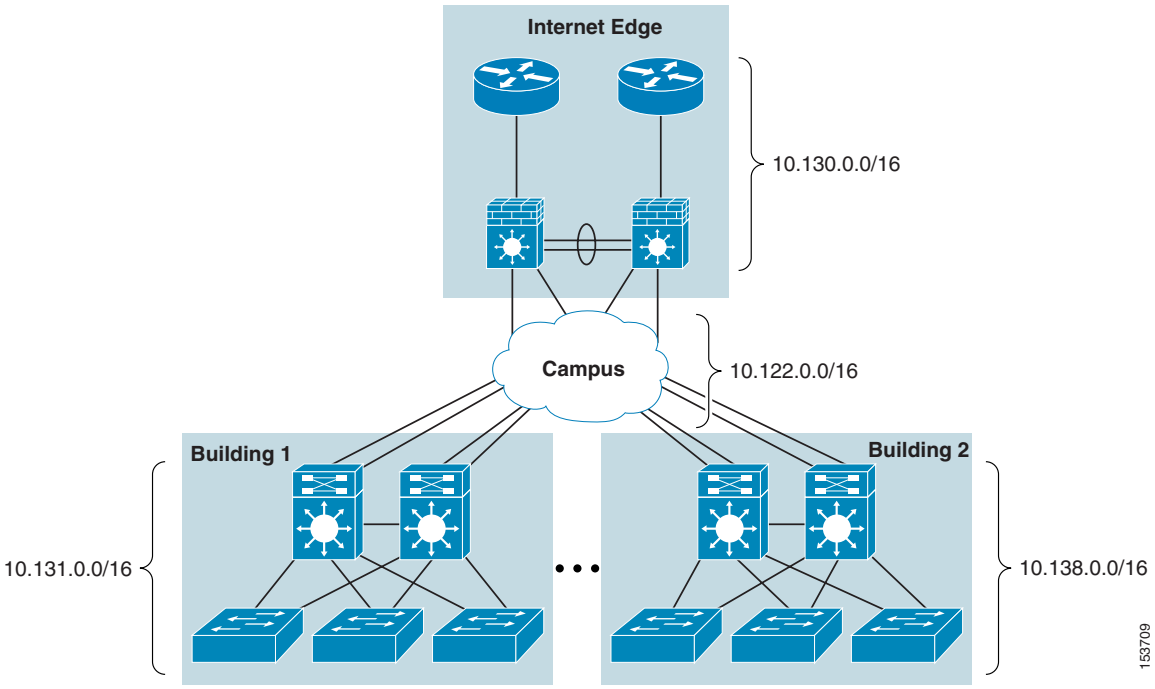
Loopback IP Address Considerations

Important design considerations arise when describing the criteria for assigning IP addresses to the loopback interfaces that represent the source and destination of the GRE tunnels. The following considerations are also valid for either point-to-point GRE or mGRE tunnel scenarios:

- The loopback interfaces must belong to the global routing table. There is currently no support on Catalyst switches for GRE source and destination interfaces belonging to a VRF.
- It must be determined from which range to take the IP addresses assigned to the loopback interfaces. The assumption here is that a proper subnet planning is in place, so that a summarized route can be used in the core from each campus building block, as shown in Figure 14.

Note The IP addresses used in this example simplify the description and are not intended to represent a best practice summarization schema.

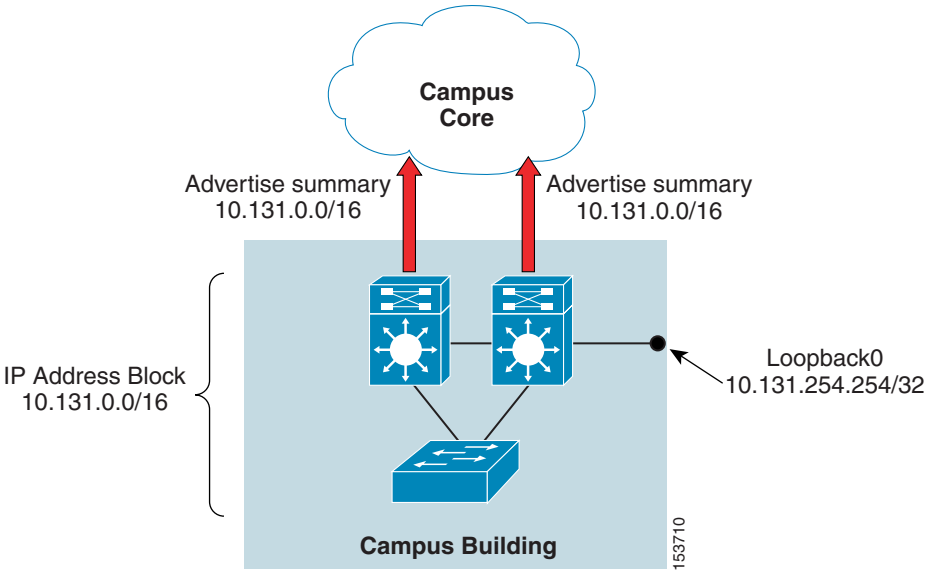
Figure 14 IP Addressing Assignment in a Campus Network



Two options of assigning IP addresses to the loopback interfaces are as follows:

- Assigning IP addresses from the valid pool in the specific distribution block where the loopbacks reside. When doing this, the specific loopback addresses is advertised to the core of the network as part of the generic network summary (for example, 10.131.0.0/16, as shown in Figure 15).

Figure 15 Assigning a Loopback Address From a Campus Building Pool



In this specific scenario, sending a network summary to the core can cause the creation of a black hole if the link between the two distribution switches fails. Because the core devices receive only the summary information, it is not possible to predict the return path for the traffic originated

elsewhere in the network and destined to any IP address that is part of that summary. In the example in Figure 15, it can happen that GRE traffic directed to the Loopback 0 on the right distribution switch is actually routed from the core to the left distribution device. At this point, it is essential to have connectivity between the two distribution switches to avoid the creation of a black hole.

When following this approach, Cisco recommends that you increase the reliability of the connections between the distribution layer peers by connecting these devices with redundant physical links (at least two) belonging to different line cards (to avoid the single point of failure represented by the switch line card itself). This can increase the cost of the solution, especially in the scenario where 10 G links are in place between the distribution switches, but it also provides the additional bandwidth required when this connection becomes a transit link.

Note Depending on the specific design, the two links might be bundled in a port channel (this is recommended when the connection is a Layer 2 trunk), or kept separate (if they are Layer 3 routed links).

- Assigning IP addresses from a different pool than the one used in the distribution block.

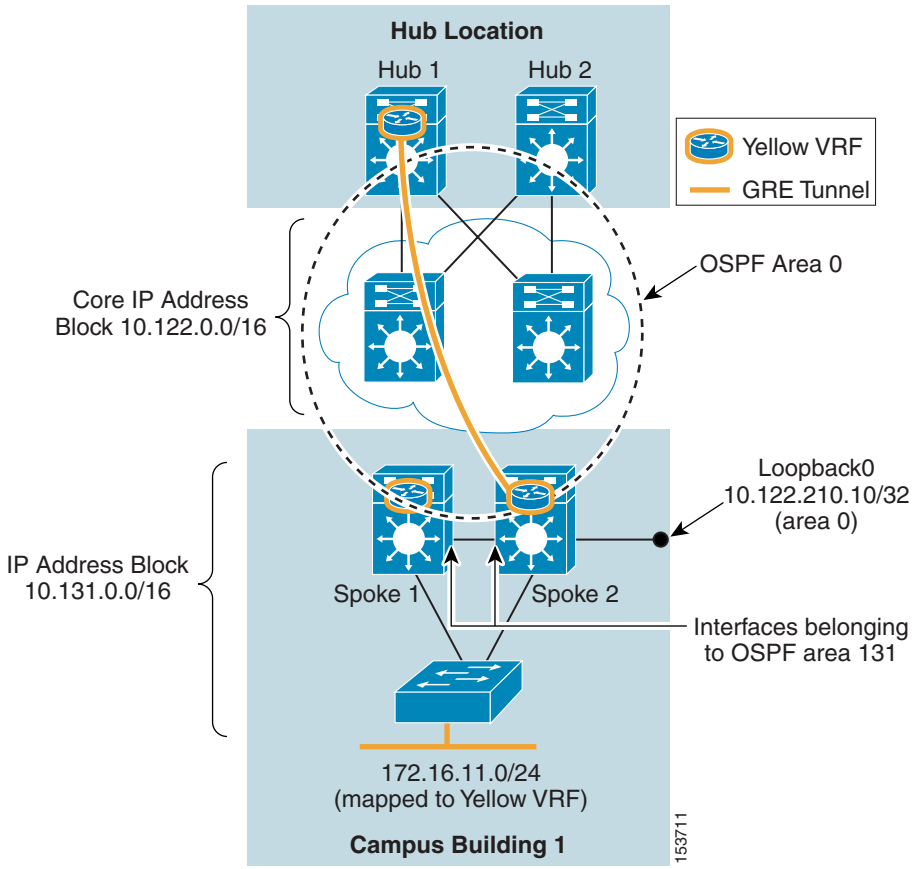
  This the recommended solution that does not present the caveat discussed above because each distribution switch advertises to the core the specific IP addresses used for the loopbacks. The drawback is that all the specific routing information for the loopbacks defined on each campus distribution block needs to be contained in the routing tables of all the other campus network devices.

  This solution presents a caveat that can lead to the creation of a black hole when the following conditions are valid:

  - OSPF is the protocol running in the global table. In this case, following the recommended guidelines to deploy campus networks, the interfaces connected to the core are configured as part of area 0, whereas the link between the distribution peers belongs to the specific area used in that block.
  - A GRE tunnel is established between the hub and the spoke to logically connect VRFs defined on the two devices. EIGRP is the routing protocol running in the context of the VRF.
  - The loopbacks have assigned IP addresses from the pool valid in the core and are configured as part of OSPF area 0.

  This scenario is shown in Figure 16.

Figure 16 Assigning a Loopback IP Address from the Core IP Address Pool



Under these circumstances, the hub device is learning from the two spokes routing information for the subnet defined in that campus distribution block and belonging to the defined VRF, as follows:

```
cr8-6500-1#sh ip route vrf yellow
Routing Table: yellow
Codes: 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, E - EGP
       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
Gateway of last resort is 172.18.3.30 to network 0.0.0.0
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.11.0 [90/15360256] via 172.32.11.3, 00:00:02, Tunnel0
                    [90/15360256] via 172.32.11.2, 00:00:02, Tunnel0
```

Note In this specific example, an mGRE interface is defined on the hub device, but the same considerations are valid when using point-to-point GRE connections.

Also, the hub device maintains valid EIGRP neighborships with both the spokes, as follows:

```
cr8-6500-1#sh ip eigrp vrf yellow neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
```

```
0   172.32.11.2             Tu0             11 00:07:20 1344  5000  0  377
1   172.32.11.3             Tu0             13 00:07:40 1147  5000  0  434
```

Now, assume that both of the connections from Spoke 2 to the core fail. This represents a single point of failure scenario in all the designs where physical ports belonging to the same switch line card are used to connect to the core (failure of the line card causes both the interfaces to go down). When this occurs, the GRE tunnel connecting the spoke to the hub becomes unidirectional. The spoke can communicate to the hub (via the peer spoke 1), but the hub cannot communicate to the spoke because it does not know how to reach the destination of the GRE tunnel (loopback 0 in this example). This happens because the loopback belongs to OSPF area 0; thus, routing information is not sent to spoke 1 because the two distribution layer devices are connected through interfaces belonging to area 131 (area 0 is partitioned).

As a result, spoke 2 brings down the EIGRP adjacency with the hub, whereas the hub that is still receiving the EIGRP hellos from spoke 2 maintains that neighborship in an active state. The direct consequence is that the hub keeps in the routing table the same routing information for the remote subnet 172.16.11.0, and this cause black-holing of traffic every time packets destined to that subnet are directed to spoke 2.

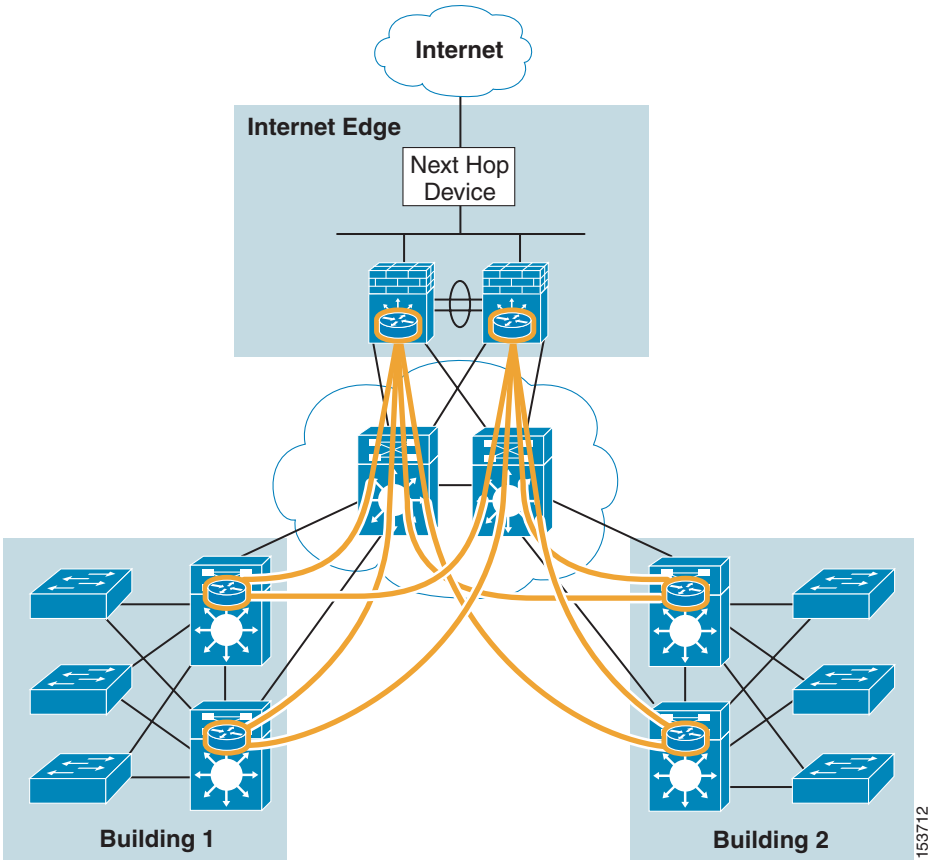
Note This problem does not happen when using OSPF in the context of the VRF, because OSPF inherently verify the existence of a two-way communication before establishing adjacency with the neighbor devices.

The recommended way to avoid this problem is to remove the single point of failure represented by the switch line card. Connections to the core from each spoke should be deployed, using physical ports belonging to separate switch modules.

High Availability Considerations

The recommended design to provide resiliency in the hub-and-spoke scenario consists of implementing redundant hub devices and creating two separate hub-and-spoke networks, connecting the spokes to each hub, as shown in Figure 17.

Figure 17 Redundant Hub-and-Spoke Overlay Networks



Each spoke device builds a separate GRE tunnel destined to the redundant pair of hubs, traffic is load balanced between the two tunnels, and each spoke learns a default route with the same metric from each hub (as described in the previous section).

Note that the overall resiliency of the overlay solution is based on the resiliency of the network infrastructure. This can be achieved by following the recommended design guidelines in the documents at the following URLs:

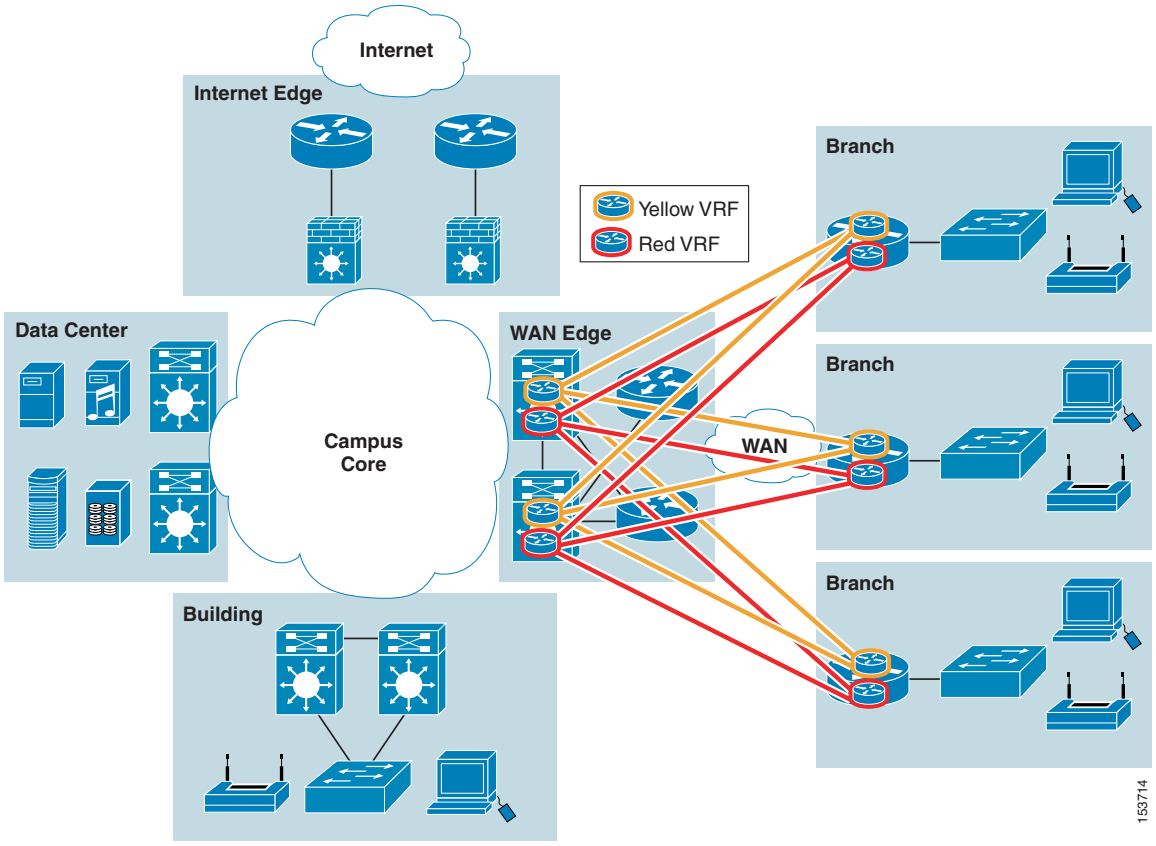
- http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/cdccont_0900aecd801a8a2d.pdf
- http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/cdccont_0900aecd801a89fc.pdf

Using VRF-Lite and GRE over the WAN

The use of VRF and GRE to provide path isolation over the WAN is recommended for all the applications requiring backhauling to the main site of the traffic originating from the remote branch locations (no branch-to-branch communication required). A typical example of such an application is providing guest access when the connection to the ISP is localized in the main site and is not available at the branch locations.

In this scenario, a hub-and-spoke overlay network can be built for each user group (VPN) defined at the branch. The hub location in this case is represented by the campus WAN edge, as shown in Figure 18.

Figure 18 Hub-and-Spoke Logical Networks Over the WAN

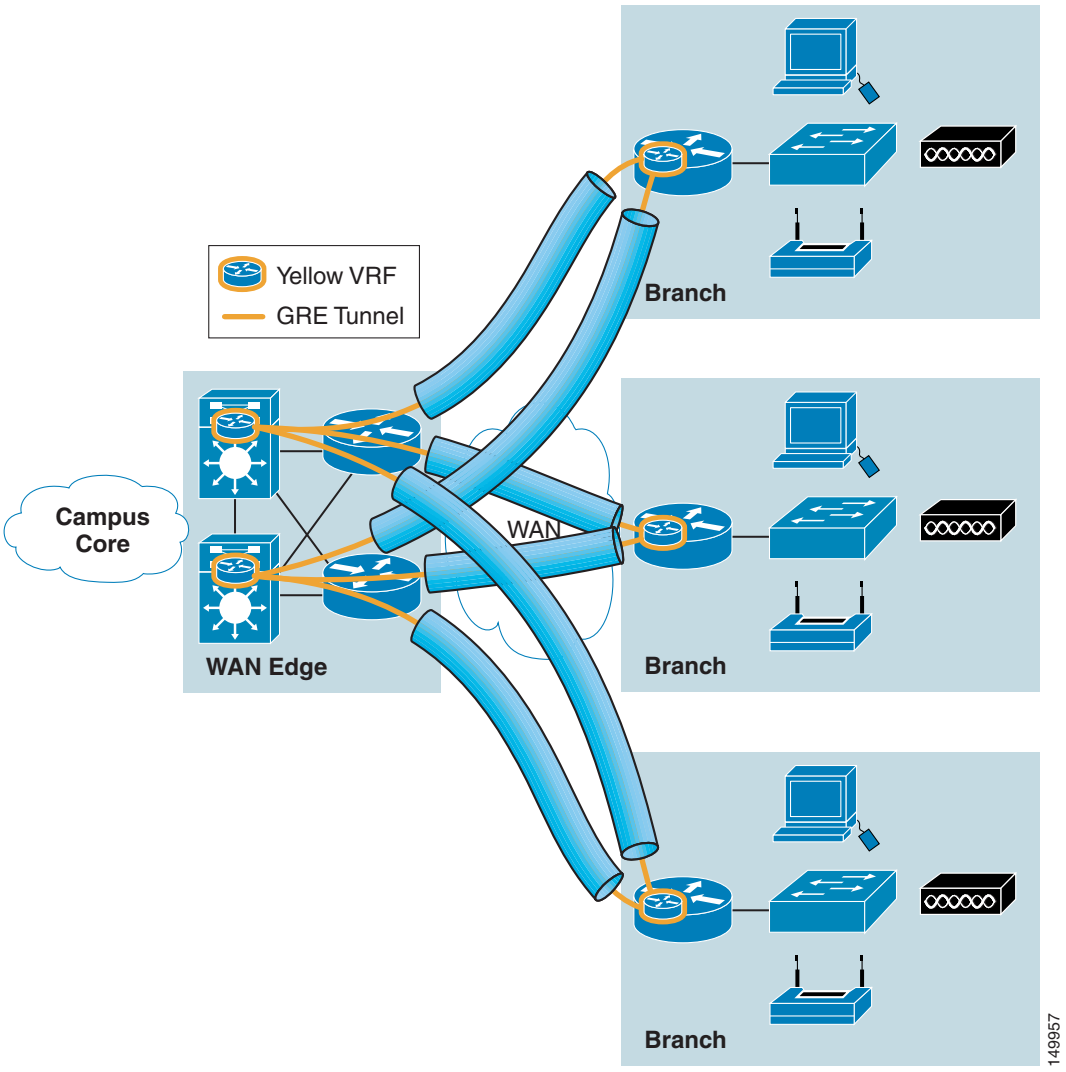


Note that a separate overlay network can be defined for each user group (VPN). In the example in Figure 18, two separate yellow and red VRFs are defined.

This design assumes a customer requirement of encrypting all the traffic sent over the WAN using IPsec. This is becoming increasingly common, even over legacy WAN clouds. A complete description of the various IPsec deployments scenarios is beyond the scope of this guide.

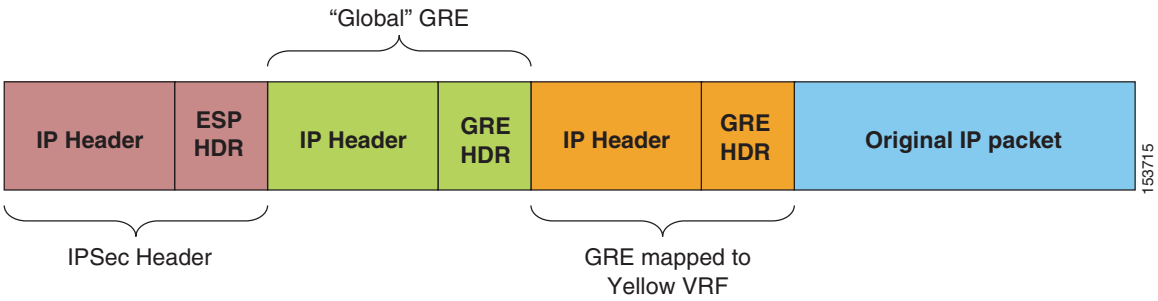
Independently from the specific type of connectivity established in the global table between the remote branch locations and the main site, these IPsec pipes are used to carry the traffic for each defined user group from each branch location to the main site in an encrypted form, as shown in Figure 19 for the yellow VPN.

Figure 19 Using IPsec Pipes as Transport



Note that if GRE is used in conjunction with IPsec to support multicast traffic and the use of dynamic routing protocol (in the global table), each IP packet sent by a user belonging to group "yellow" in the VRF is sent over the WAN in the format shown in Figure 20.

Figure 20 IP Packet for User in Yellow Group Over the WAN



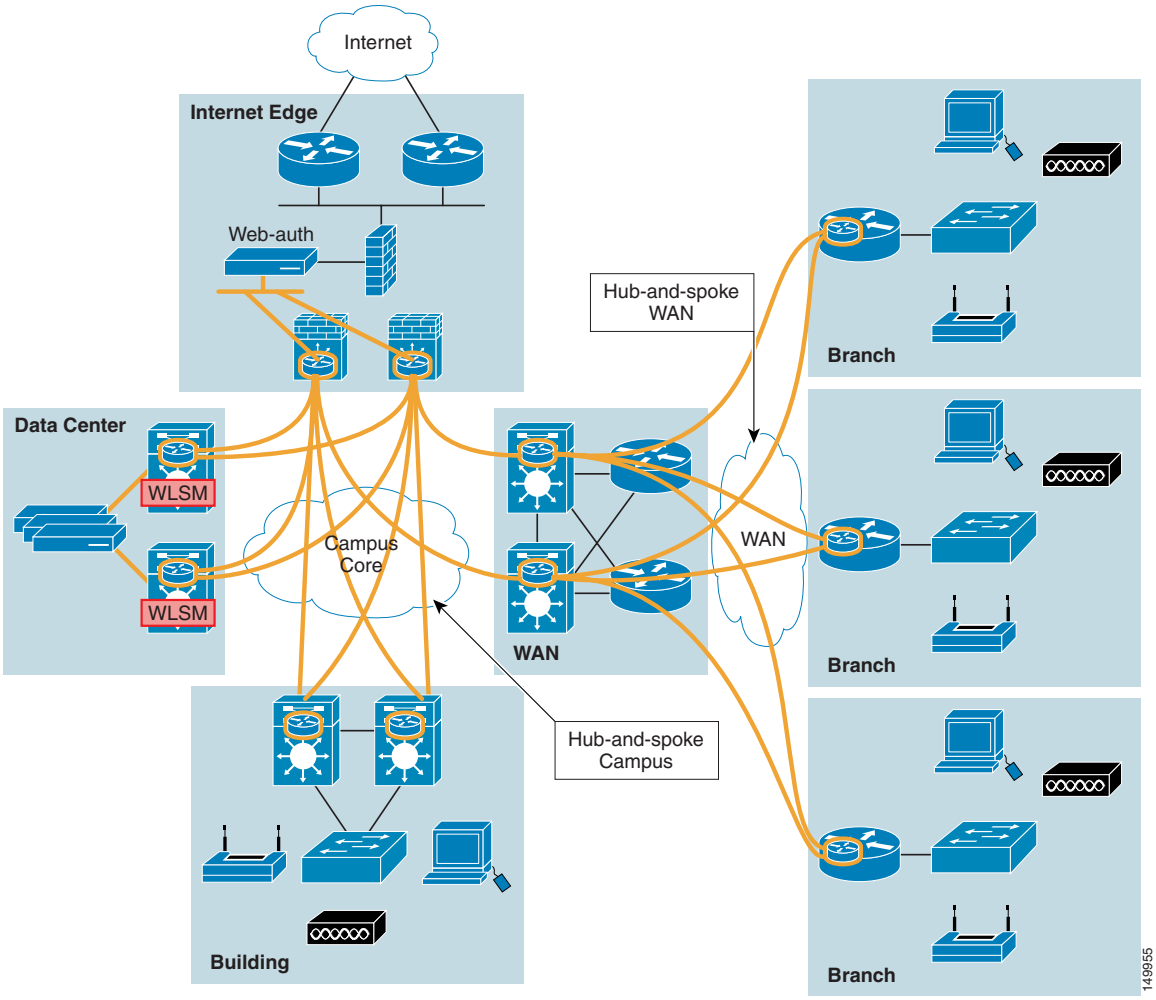
Basically, three additional headers are imposed on the original IP packet originated from the yellow endpoint. In this design, these different headers can be stripped on different devices in the campus WAN edge block, according to the following sequence of events:

1. The branch router receives the original IP packet on the interface mapped to the yellow VRF.
2. The branch router adds the three headers to the original packet and sends it into the WAN cloud.
3. The packet is received on the campus WAN edge block by one of the routers facing the WAN cloud (for example, a Cisco 7200 router platform). The traffic is decrypted, so the external IPsec header is removed.
4. After decryption, the packet is handed to the tunnel interface defined in global table on the WAN router and is GRE decapsulated.
5. Up to this point, traffic belonging to a specific user group (belonging to a VRF) or global traffic (handled in global routing table) is treated in the same way. The difference now is that traffic originated from a remote subnet mapped to the yellow VRF, for example, presents an additional GRE header, so when the packet is sent from the WAN router to the switch facing the campus core (a Catalyst 6500 in this design), it is handed to the tunnel interface that is mapped to the corresponding VRF.
6. At this point, the original IP packet is routed in the context of the yellow VRF and does not use the information in the global routing table or in other VRFs, which is the goal.

Because of the overhead caused by the additional headers being added to the original IP packets, it is important to manually lower the MTU on the GRE tunnel interface that is mapped to each VRF. The same considerations described in High Availability Considerations, page 43 are also valid here because the goal is still to use PMTUD to avoid fragmentation of the IP packets. To accommodate all the additional headers shown in Figure 20, a suggested MTU value of 1300 can be used (see Configuration Details, page 15).

After the traffic is received on the Catalyst 6500 switches in the WAN edge distribution, it is sent toward its destination using the path isolation solution in the campus network. Figure 21 shows an example of a guest access solution.

Figure 21 Campus and WAN Network Virtualization for Guest Access



In this specific example, two distinct logical overlay networks have been built; one to handle the guest traffic inside the campus network, the second for carrying it over the WAN. The two Catalyst 6500 switches positioned in the campus WAN edge perform a dual role:

- The switches represent the hubs for the overlay logical network built over the private WAN to aggregate the guest traffic originated from all the remote branch locations.
- The switches represent the spoke devices for the logical network built across the campus core. Their function in this case is to forward all the guest traffic toward the campus Internet edge devices.

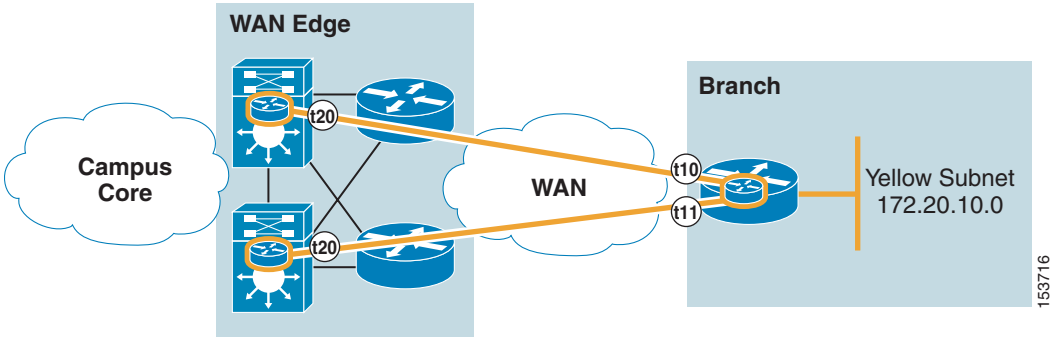
The example in Figure 21 illustrates the same path isolation solution (hub-and-spoke achieved with VRF and GRE) used in the campus network and also over the WAN. This design is not mandatory because any good design should maintain independence between the two logical overlay networks. Terminating the GRE tunnels originated at the branch locations in the WAN distribution block provides a complete separation between the network virtualization solution adopted inside the campus and over the WAN. For example, if a decision is made to deploy Multiprotocol Label Switching (MPLS) in the campus core, a specific MPLS VPN can be dedicated to handle the guest traffic. The Catalyst 6500 in the WAN block is then the PE device for the MPLS VPN campus deployment, and is used to send all the traffic received over the GRE tunnels into the MPLS VPN realm.

The recommended design imposes some restrictions on the choice of the Catalyst platforms that can be deployed in the WAN distribution block. To aggregate the GRE tunnels originated at the branch locations, Cisco recommends that you use Catalyst 6500 switches with Sup720 because of their ability to switch GRE traffic in hardware. This is basically the same requirement as is given for the Internet edge. For more information on GRE implementation for Catalyst 6500 equipped with Sup720, see the following URL: http://bock-bock.cisco.com/~csolder/

Configuration Details

The following configuration samples are used to create the overlay network on the WAN, distinguishing the commands required on the Catalyst 6500 switches in the distribution layer of the WAN block and on the branch routers. (See Figure 22.)

Figure 22 Deploying the Overlay Network On the Private WAN



Catalyst 6500 WAN

The Catalyst 6500 switches in the WAN block have a dual role. They represent the redundant headend for the hub-and-spoke network connecting to the branch offices, and they also provide the point of entrance to the network virtualization solution deployed inside the campus. The recommended configuration for building the overlay logical network extending over the WAN uses mGRE interfaces on the headend devices, because of the advantages described in Using mGRE Technology, page 32 for a campus environment. The required configuration steps for the Catalyst 6500 platforms in the WAN edge are described as follows. (This is also valid for a generic yellow VRF.)

Step 1 Define the yellow VRF:

```
ip vrf yellow
 rd 100:1
```

Step 2 Define the mGRE interface to aggregate the guest traffic originated at the remote locations:

```
interface Loopback20
 description source mGRE for branches
 ip address 10.127.200.1 255.255.255.255
!
interface Tunnel20
 description mGRE hub for branches
 ip vrf forwarding yellow
 ip address 172.20.1.1 255.255.255.0
 no ip redirects
 ip mtu 1376
 ip nhrp map multicast dynamic
```

```
 ip nhrp network-id 100
 tunnel source Loopback20
 tunnel mode gre multipoint
```

Step 3 Enable a routing protocol in the context of the yellow VRF, adding the proper route filtering, as previously described in the campus scenario. The following configuration sample is used for an EIGRP deployment. For an OSPF configuration example, see Enabling a Routing Protocol, page 27.

```
router eigrp 100
 passive-interface default
 no passive-interface Tunnel10
 no passive-interface Tunnel11
 no passive-interface Tunnel20
 no auto-summary
 !
 address-family ipv4 vrf yellow
 network 172.32.10.0 0.0.0.255
 network 172.32.11.0 0.0.0.255
 network 172.20.1.0 0.0.0.255
 distribute-list import_routes in
 no auto-summary
 autonomous-system 100
 exit-address-family
!
ip access-list standard import_routes
 permit 172.20.10.0 0.0.0.255
```

Note The last statement in this example (permit 172.20.10.0 0.0.0.255) is required to allow the yellow subnets defined at the remote branch location in routing table. In a real deployment, there can be more than one of these statements; one for each remotely defined yellow subnet.

Branch Router

The following configuration sample is valid for the branch router and applies to the network diagram in Figure 22.

Step 1 Define the yellow VRF:

```
ip vrf yellow
 rd 100:1
```

Step 2 Define the dual logical connection to the WAN edge. Note that these GRE tunnel interfaces are point-to-point because it is assumed that there is no need for spoke-to-spoke communication.

```
interface Loopback10
 description source GRE to WAN hub 1
 ip address 10.127.210.1 255.255.255.255
!
interface Loopback11
 description source GRE to WAN hub 2
 ip address 10.127.211.1 255.255.255.255
!
interface Tunnel10
 description GRE tunnel to WAN hub 1
 ip vrf forwarding yellow
 ip address 172.20.1.2 255.255.255.0
 ip mtu 1376
```

```
 ip nhrp network-id 20
 ip nhrp nhs 172.20.1.1
 ip nhrp registration timeout 60
 tunnel source Loopback10
 tunnel destination 10.127.200.1
!
interface Tunnel11
 description GRE tunnel to WAN hub 2
 ip vrf forwarding yellow
 ip address 172.20.2.2 255.255.255.0
 ip mtu 1376
 ip nhrp network-id 20
 ip nhrp nhs 172.20.2.1
 ip nhrp registration timeout 60
 tunnel source Loopback11
 tunnel destination 10.127.201.1
```

Step 3 Define the yellow subnet for the branch location (in a real scenario, there can be multiple subnets defined at each branch location).

```
interface FastEthernet0/1.20
 description Branch Guest subnet
 encapsulation dot1Q 20
 ip vrf forwarding yellow
 ip address 172.20.10.1 255.255.255.0
 ip helper-address 172.18.2.10
    "Enable routing.
router eigrp 100
 passive-interface default
 no passive-interface FastEthernet0/1.20
 no passive-interface Tunnel10
 no passive-interface Tunnel11
 no auto-summary
 !
 address-family ipv4 vrf yellow
 network 172.20.1.0 0.0.0.255
 network 172.20.2.0 0.0.0.255
 network 172.20.10.0 0.0.0.255
 distribute-list import_routes in
 no auto-summary
 autonomous-system 100
 exit-address-family
!
ip access-list standard import_routes
permit 0.0.0.0
```

Note The distribute-list statement is required to install a default route that points to the hub device in the WAN edge in routing table.

QoS in Hub-and-Spoke Deployments

Congestion inside a campus network is a rare event during normal operating conditions because of the large amount of available bandwidth. However, during an abnormal event, such as denial-of-service (DoS) or worm attacks, campus congestion can typically occur within minutes (even in 10 GE networks), as part of the collateral damage of such an attack. Therefore, classification and metering of traffic at the edge of the network is a valuable worm mitigation strategy. This strategy is even more relevant in

scenarios where the enterprise does not usually have any control over the connected guest machines and therefore cannot enforce any security policies; for example, as when providing guest access. However, at enterprise branch locations, classification and metering of traffic becomes a priority, to achieve proper use of the bandwidth resources available across the WAN cloud.

The scenario described in this section relates to providing QoS for applications requiring hub-and-spoke connectivity; this is very relevant for GRE and VRF deployments and for the specific business problems they aim to solve (for example, guest access or for NAC remediation designs).

Two approaches are described in this section for classifying and handling traffic in hub-and-spoke deployments. The assumption is that there is the need to somehow limit the traffic originated from the edge of the network; for example, this is valid in guest access deployments.

The first approach strictly rate-limits the traffic originated from the subnets at the edge of the network, so that traffic exceeding a predetermined threshold is dropped and not allowed into the core of the network. The exact value to be used for the thresholds can vary from design to design. The goal of this section is to provide all the required tools to rate-limit traffic for both wired and wireless deployments. The main advantage of this approach is its simplicity, and also that its functionality is independent from the deployment of end-to-end QoS across the network.

The second approach is more dynamic and classifies the traffic at the edge and then prioritizes it inside the network. Again, the recommended strategy for classifying and marking the traffic is based on the definition of a specific threshold. Traffic within the threshold is treated as good faith, best effort traffic. Traffic exceeding the given allowance is marked as scavenger traffic and is aggressively dropped in the event of congestion. For these configuration examples, a threshold of 1 Mbps is used. However, note that this is just a sample value used for the configuration samples in this guide (the value of this threshold likely varies from enterprise to enterprise). Specific to these examples, traffic up to 1 Mbps is marked as best effort traffic (DSCP 0), whereas traffic that exceeds the threshold is marked as scavenger traffic (CS1 or DSCP 8).

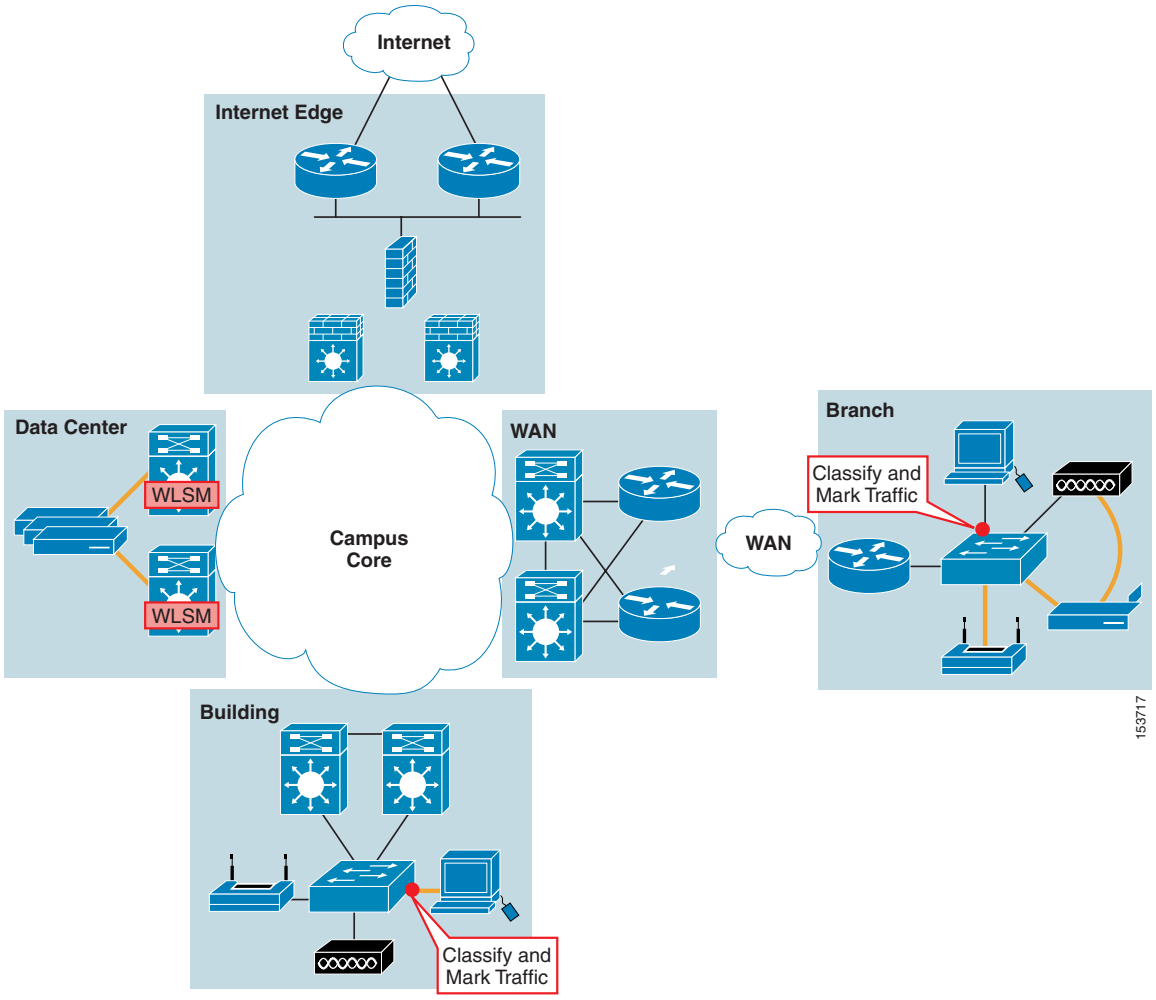
The scavenger class of traffic was introduced to offer a less than best-effort service. Access layer policers mark out-of-profile traffic to CS1/DSCP8 (scavenger), and then have all congestion management policies provision a less than best-effort queuing service for this type of traffic. Traffic marked as scavenger starts being aggressively dropped whenever congestion occurs on campus or WAN edge links. If no congestion is experienced, the available bandwidth is successfully used. An approach based on the use of scavenger-class QoS is much more flexible and dynamic than a strict rate-limiting of traffic at the edge of the network. Additionally, it provides a worm mitigation strategy in cases where clients connected to the enterprise network become infected by a virus and the virus starts attacking the network infrastructure. To be effective, it is assumed that all the devices in the network have been configured with the proper Differentiated Services Code Point (DSCP) trust boundaries and queuing and dropping strategies. The configuration details to achieve this prioritization are beyond the scope of this guide. For more information on how to accomplish this for both campus and WAN scenarios, see the *Enterprise QoS Solution Reference Network Design Guide* at the following URL: http://www.cisco.com/univercd/cc/td/doc/solution/esm/qossrnd.pdf.

The following sections describe how to configure the devices at the edge of the network, indicating the various configuration steps that are required on various platforms for both of these approaches.

Wired Clients

Traffic originated from wired clients is received on the access layer switches deployed in each campus building block or at each branch location. Classification and marking should be applied to these devices, as shown in Figure 23.

Figure 23 Classifying and Marking Traffic for Wired Clients



The most granular policing can be achieved by using per-port/per-VLAN policers that are supported on the Catalyst 2970, 3560, 3750, and 4500. Using per-port/per-VLAN policing has the following advantages:

- It defines a generic policy that is portable and that can be seamlessly applied across various access layer devices.
- It applies the policy to a physical port, and the policy is effective only when that port is deployed in the specified VLAN. This is important in a design where the same switch port is dynamically assigned to a different VLAN, based on the identity of the connected user.

For Catalyst 6500 switches, a different approach is used, given the lack of per-port/per-VLAN policing. See Catalyst 6500, page 56 for more information.

Note When deploying the static approach for wired clients, the recommended design consists of creating a two-tiered policy. At the access layer, traffic is rate-limited per port (per user) up to a certain threshold. At the distribution layer, an ingress policer is configured on the trunk ports connecting to the access layer devices, so that the aggregate traffic can be rate limited. The required configuration commands to create this two-tiered policy depend on the specific platforms deployed at the access and distribution layers.

Catalyst 2970, 3560, and 3750

Per-port/per-VLAN policing requires Cisco IOS Release 12.2(25)SE or later. To enable this functionality, you must use hierarchical policy maps. The required configuration steps follow. Most of the commands are common for both the static and dynamic approaches. The only difference is in the creation of the interface-level policy map.

Step 1 Enable QoS globally:

```
3560-Access(config)#mls qos
```

Step 2 Enable VLAN-based QoS on the switch port.

By default, VLAN-based QoS is disabled on all physical switch ports. The switch applies QoS, including class maps and policy maps, only on a physical port basis. In Cisco IOS Release 12.2(25)SE or later, you can enable VLAN-based QoS on a switch port. This procedure is required on physical ports that are specified in the interface level of a hierarchical policy map on an SVI (defined in the next step).

```
3560-Access(config)#int f0/17
3560-Access(config-if)#mls qos vlan-based
```

Step 3 Configure hierarchical policing.

Hierarchical policing combines VLAN and interface level policy maps to create a single policy map. On an SVI, the VLAN-level policy map specifies on which traffic class to act. Actions can include trusting the class of service (CoS), DSCP, or IP precedence values, or setting a specific DSCP or IP precedence value in the traffic class. The following steps are required for marking the traffic originated in a generic edge VLAN, accordingly to the strategy previously described.

a. Create a VLAN-level class map. Note that the ACL is generically defined to match all the IP traffic. This is the key for the ACL portability previously mentioned.

```
3560-Access(config)#access-list 101 permit ip any any
3560-Access(config)#class-map match-all EDGE-VLAN
3560-Access(config-cmap)#match access-group 101
```

b. Create an interface-level class map to specify the physical switch ports that are affected by the policer.

```
3560-Access(config)#class-map match-all EDGE-INTF
3560-Access(config-cmap)#match input-interface f0/1 - f0/48
```


Note You can specify all the switch ports in the **match input-interface** command. The policer works on a given switch port only if it is part of the specified VLAN.

c. Create an interface-level policy map to define the action to take on traffic received on each port.

- Static approach

  Traffic exceeding a specified threshold is dropped. Traffic below the threshold is marked as best effort and is transmitted.

```
3560-Access(config)#policy-map EDGE-INTF-POLICY
3560-Access(config-pmap)#class EDGE-INTF
3560-Access(config-pmap-c)#set dscp default
3560-Access(config-pmap-c)#police 1000000 8000 exceed-action drop
```

- Dynamic approach

  In this case, all the traffic that exceeds a specified threshold (1 Mbps in the example) is marked as scavenger traffic, but is not dropped.

```
3560-Access(config)#mls qos map policed-dscp  0 to 8
3560-Access(config)#policy-map EDGE-INTF-POLICY
3560-Access(config-pmap)#class EDGE-INTF
3560-Access(config-pmap-c)#police 1000000 8000 exceed-action policed-dscp-transmit
```

d. Create the VLAN-level policy map:

```
3560-Access(config-pmap)#policy-map EDGE-VLAN-POLICY
3560-Access(config-pmap)#class EDGE-VLAN
3560-Access(config-pmap-c)#set dscp default
3560-Access(config-pmap-c)#service-policy EDGE-INTF-POLICY
```

e. Apply the previously define policy map to the SVI. This is the key step to ensure that the policer is effective on switch ports belonging to this VLAN (and only on these).

```
3560-Access(config)#interface vlan 21
3560-Access(config-if)#service-policy input EDGE-VLAN-POLICY
```

Catalyst 4500

The configuration of per-port/per-VLAN policing on Catalyst 4500 platforms is more straightforward than for the Catalyst 2970, 3560, and 3750, because it does not require the definition of hierarchical policy maps. To support this functionality, Cisco IOS Release 12.2(25)EWA or later is required (for Sup2+ to Sup V). The required configuration steps follow. Most of the commands are common for both the static and dynamic approaches, The only difference is in the creation of the policy map.

Step 1 Create a class map to identify the traffic:

```
4500-Access(config)#access-list 101 permit ip any any
4500-Access(config)#class-map match-all EDGE-VLAN
4500-Access(config-cmap)#match access-group 101
```

Step 2 Define the policy map to mark the traffic.

- Static approach

  Traffic exceeding a specified threshold should be dropped, whereas traffic below the threshold is marked as best effort and is transmitted.

```
4500-Access(config)#policy-map EDGE-VLAN-POLICY
4500-Access(config-pmap)#class EDGE-VLAN
4500-Access(config-pmap-c)#set ip dscp 0
4500-Access(config-pmap-c)#police 1000000 8000 exceed-action drop
```

- Dynamic approach

  In this case, all traffic exceeding a specified threshold (1 Mbps in the example) is marked as scavenger traffic but is not dropped.

```
4500-Access(config)#qos map dscp policed 0 to dscp 8
4500-Access(config)#policy-map EDGE-VLAN-POLICY
4500-Access(config-pmap)#class EDGE-VLAN
4500-Access(config-pmap-c)#set ip dscp 0
4500-Access(config-pmap-c)#police 1000000 8000 exceed-action policed-dscp-transmit
```

Step 3 Apply the policy map.

Note that this is done on a per-VLAN basis on each physical interface. This means that the policy is in effect only when the port is configured as part of that VLAN (or if it is a trunk carrying that VLAN).

```
cr24-4503-1(config)#int g2/1
```

```
cr24-4503-1(config-if)#vlan-range 11
cr24-4503-1(config-if-vlan-range)#service-policy input EDGE-VLAN-POLICY
```

Catalyst 6500

The Catalyst 6500 is the most powerful and flexible Cisco switching platform. As such, it can be found in all three layers of a campus network (access, distribution, and core). When configured as an access layer switch, traditionally the software running on the Supervisor is CatOS. When configured as a distribution or core layer switch, the recommended software is Cisco IOS. This distinction has changed since the introduction of the Sup32, which can run both CatOS and IOS code and is usually positioned as an access layer device. See Catalyst 6500 with Cisco IOS, page 57 for the Cisco IOS configuration.

Note In this section, only Catalyst 6500 Supervisors equipped with Policy Feature Card 2 (PFC2) or PFC3 are taken into consideration. This categorization includes Sup2 (PFC2) and Sup32/Sup720 (PFC3), but not older Supervisor models (Sup1/Sup1a).

Catalyst 6500 with CatOS

Per-VLAN policers are supported in CatOS. However, this type of policer should not be confused with the per-port/per-VLAN policers described in the previous sections for the other Catalyst platforms.

A per-VLAN policer can police all flows within a given VLAN, as an aggregate sum of the traffic of all ports belonging to a given VLAN. A per-port/per-VLAN policer can discretely police flows from a given VLAN on a per-port basis, which is much more granular than other policing methods. Because the purpose of the design described here is to classify and mark the traffic received on each switch port, the aggregate per-VLAN policer is not used in this example; a port-based QoS is configured instead.

The required configuration steps follow. Most of the commands are common for both the static and dynamic approaches. The only difference is in the definition of the aggregate policer.

Step 1 Define the aggregate policer to be used for the edge traffic.

When configuring per-port policers in CatOS, a default behavior to keep in mind is that, in CatOS, ACLs and aggregate policers cannot be applied to more than one port at the same time. For example, if an aggregate policer called POLICE-EDGE is defined to rate-limit flows to 1 Mbps, and this policer is applied to two separate ports in CatOS, it rate-limits flows from both ports to a *combined* total of 1 Mbps, instead of the intended behavior of limiting flows to 1 Mbps on a per-port basis (as is the case if configured in Cisco IOS). To work around this default behavior, ACLs and aggregate policers have to be uniquely defined on a per-port basis.

- Static approach

  Traffic exceeding a specified threshold is dropped, whereas traffic below the threshold is marked as best effort and is transmitted.

  ```
  6500-access> (enable) set qos policer aggregate EDGE-PORT-2-1 rate 1000 burst 8000
  drop
  ```

- Dynamic approach

  In this case, all the traffic exceeding a specified threshold (1 Mbps in the example) is marked as scavenger traffic but is not dropped.

  ```
  6500-access> (enable) set qos policed-dscp-map 0:8
  6500-access> (enable) set qos policer aggregate EDGE-PORT-2-1 rate 1000 burst 8000
  policed-dscp
  ```

```
    "Bind an ACL to the policer to mark in-profile traffic as Best Effort (DSCP 0).
6500-access> (enable) set qos acl ip EDGE-ACL-2-1 dscp 0
aggregate EDGE-PORT-2-1 ip 10.124.10.0 0.0.0.255 any
```

Note Because the policy is applied to the physical switch ports, you need to take into consideration the fact that the same port can be used by different categories of users. For this reason, you need to define a more specific ACL to select the IP subnets from where the traffic originates. As a result, you lose the advantage of having a generic template seamlessly valid on different edge devices (which is possible when using the per-port/per-VLAN functionality, as previously described).

Step 2 Commit the ACL to PFC hardware:

```
6500-access> (enable) commit qos acl EDGE-ACL-2-1
```

Step 3 Attach the ACL to the corresponding switch port:

```
6500-access> (enable) set qos acl map EDGE-ACL-2-1 2/1
```

Catalyst 6500 with Cisco IOS

Hardware advancements in the PFC3 provide a number of new features, such as User-Based Rate Limiting (UBRL). UBRL is a form of microflow policing that provides rate-limited traffic flows and, unlike a normal microflow policer, it allows a policer to be applied to all traffic to or from a specific user.

In this section, UBRL is used to classify and mark the edge traffic. Each flow is examined by its source IP address and if a source is transmitting out-of-profile, the excess traffic can be dropped or marked as scavenger traffic (CS1 or DSCP 8), depending on the adopted approach.

The definition of a flow is determined by the flow mask; the flow mask is what defines a flow. The flow mask identifies fields in the packet header that are used to perform a lookup in the NetFlow table. In this case, use the source-only flow mask. The PFC maintains one entry for each source IP address, so that all flows from the given source IP address use this entry.

The configuration steps follow. Most of the commands are common for both the static and dynamic approaches. The only difference is in the definition of the policy map.

Step 1 Define the class-map to identify the edge traffic:

```
6500-access(config)#access-list 101 permit ip 172.16.11.0 0.0.0.255 any
6500-access(config)#class-map match-all EDGE
6500-access(config-cmap)#match access-group 101
```

Step 2 Define the policy map. It is important to specify **mask src-only** in the **police flow** command to police all the traffic sent by each specific user. To do that, configure a null flow mask for NDE (NetFlow) using the **no mls flow ip** command (this is the default value for Sup720/Sup32).

- Static approach

  Traffic exceeding a specified threshold is dropped, whereas traffic below the threshold is marked as best effort and is transmitted.

```
6500-access(config-cmap)#policy-map EDGE-POLICING
6500-access(config-pmap-c)#class EDGE
6500-access(config-pmap-c)#set dscp default
6500-access(config-pmap-c)#police flow mask src-only 1000000 8000 conform-action
transmit exceed-action drop
```

- Dynamic approach

  In this case, all the traffic exceeding a specified threshold (1 Mbps in the example) is marked as scavenger traffic but is not dropped.

```
6500-access(config)#mls qos map policed-dscp normal 0 to 8
6500-access(config-cmap)#policy-map EDGE-POLICING
6500-access(config-pmap-c)#class EDGE
6500-access(config-pmap-c)#set dscp default
6500-access(config-pmap-c)#police flow mask src-only 1000000 8000 conform-action
transmit exceed-action policed-dscp-transmit
```

Step 3 Attach the policy map to the physical interfaces:

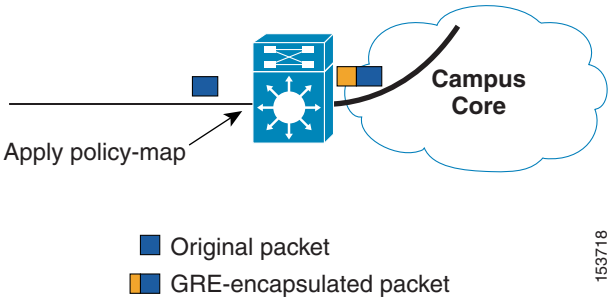
```
6500-access(config)#interface GigabitEthernet1/1
6500-access(config-if-range)#service-policy input EDGE-POLICING
```

Note In cases where the policy map is attached to a VLAN interface instead of to a physical port, you must also use the **mls qos vlan-based** command on the switch port (belonging to that specific VLAN) where the traffic is received, as shown in the following example.

```
6500-access(config)#interface GigabitEthernet1/14
6500-access(config-if)#sw acc vlan 100
6500-access(config-if)#mls qos vlan-based
6500-access(config-if)#interface vlan 100
6500-access(config-if)#service-policy input EDGE-POLICING
```

Whenever an inbound policy map is applied to a physical or logical interface of a Catalyst 6500 with PFC3, the DSCP is set on the ASIC of the egress line card before sending out the packet. This has an important consequence when the traffic needs to be sent on a tunnel interface (see Figure 24).

Figure 24 Applying a Policy Map Before Tunneling Traffic



Because of this hardware functionality, the DCSP field is set correctly in the outer IP header but not in the original IP header. This needs to be taken into consideration when the traffic is decapsulated on the switch terminating the GRE tunnel because, at that point, the marking information is no longer available.

Note This problem does not exist when traffic is not encapsulated because, in that case, only one IP header is present.

Wireless Clients

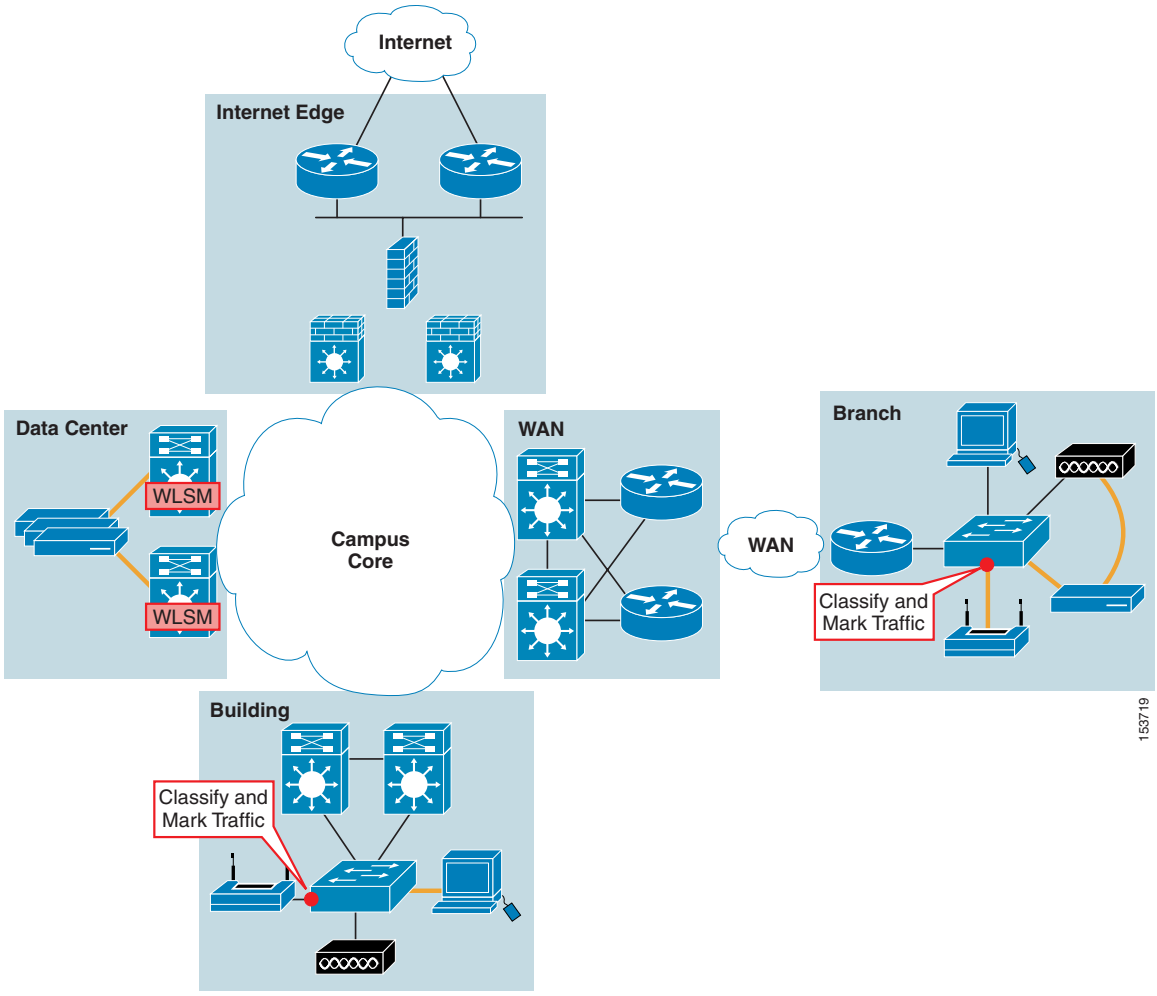
Marking strategies for traffic originating from wireless clients vary with the specific wireless deployment and with the network location (campus or branch). The same marking strategies described in the previous sections can also be applied for wireless deployments. The main difference is that now marking cannot be done on a user basis (as is done in the wired case using the per-port/per-VLAN functionality), but is done more on an aggregate basis, as described in the following sections.

As previously described for a wired scenario, a static and a dynamic QoS approach also applies for wireless deployments.

Traditional Aironet

When deploying standalone access points at the edge of the network, the traffic originating from wireless clients is locally bridged to a VLAN defined on the access layer and distribution layer network devices. This situation is identical to the wired case previously described, so the classification and marking strategies described in the previous sections can be implemented on the access layer port where the access points are connected. This is valid for both campus and branch deployments, as shown in Figure 25.

Figure 25 Classifying and Marking Traffic in a Traditional Wireless Deployment



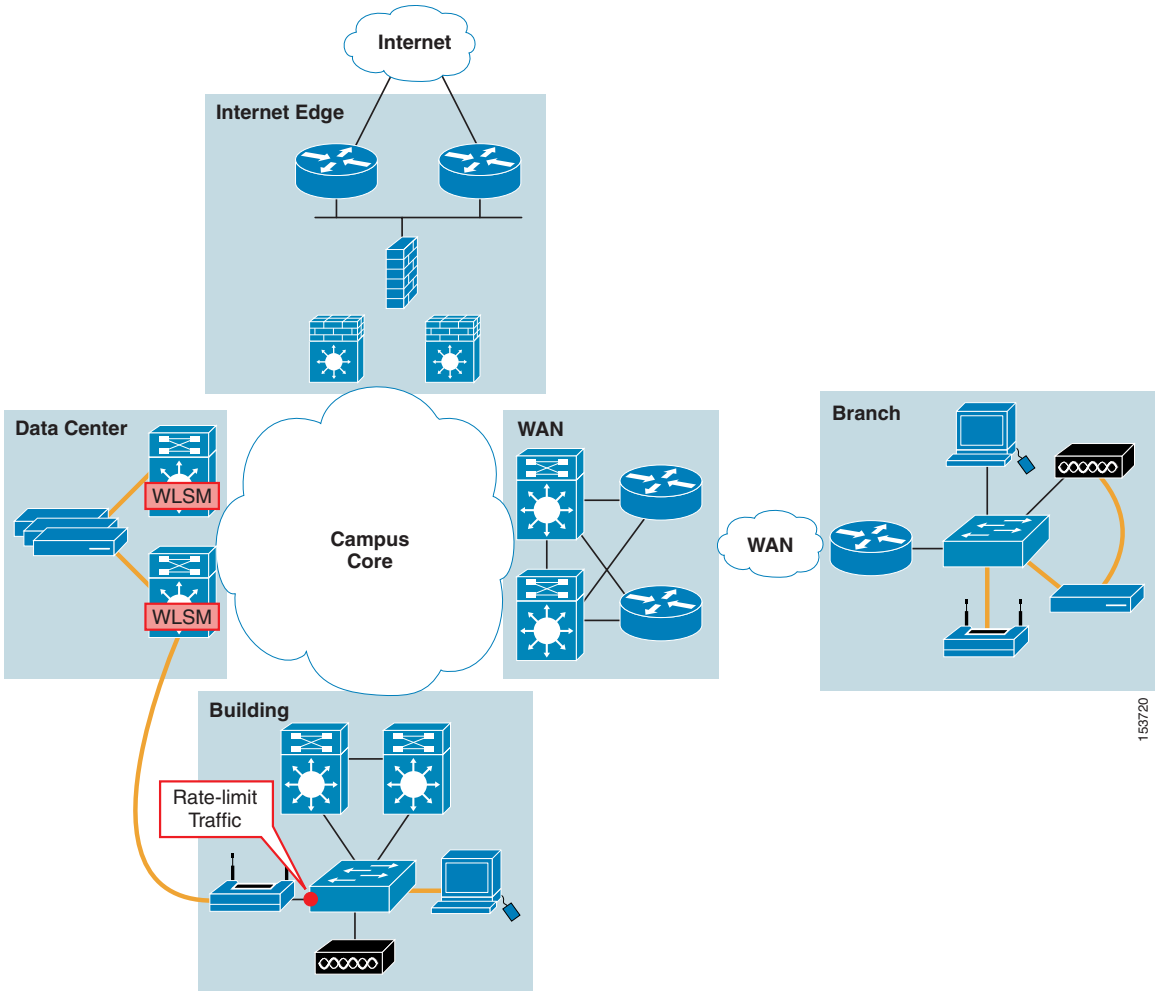
WLSM

In a wireless deployment using WLSM, the traffic is GRE-encapsulated on the access points distributed at the edge of the network and is then conveyed to a central location where the WLSM is located (in this example, this is in the enterprise data center). As a result, there are two kinds of traffic to consider: GRE traffic originated on the edge access points and directed to the Catalyst 6500 equipped with WLSM, and decapsulated traffic entering the wired portion of the network at the same Catalyst 6500 switch.

Static Approach

As previously mentioned, when deploying the static approach, the idea is to strictly rate-limit the traffic at the edge of the network. Traffic exceeding the predefined threshold is dropped and is not allowed further into the network. As a result, even for WLSM deployments, Cisco recommends performing ingress policing on the access layer switches, as shown in Figure 26.

Figure 26 Rate-Limiting Traffic on the Access Layer Device



Note that the per-port/per-VLAN functionality does not help much in this case because all the GRE traffic is sent out on the same VLAN (access point management VLAN) regardless of to which SSID (user group) the clients belong. To statically rate-limit the traffic for a specific user group, you must configure an ACL matching the destination address of the GRE tunnel that originated on the AP and associate it to the corresponding SSID. This still allows for the creation of a generic ACL that can be applied across different access layer devices.

Following is a sample configuration that is valid for a Catalyst 3560, and easily extendable to other Catalyst platforms:

Step 1 Define the class map to identify the edge traffic:

```
3560-access(config)#access-list 110 permit gre any host 10.121.253.254
3560-access(config)#class-map match-all EDGE-GRE
3560-access(config-cmap)#match access-group 110
```

Step 2 Define the policer:

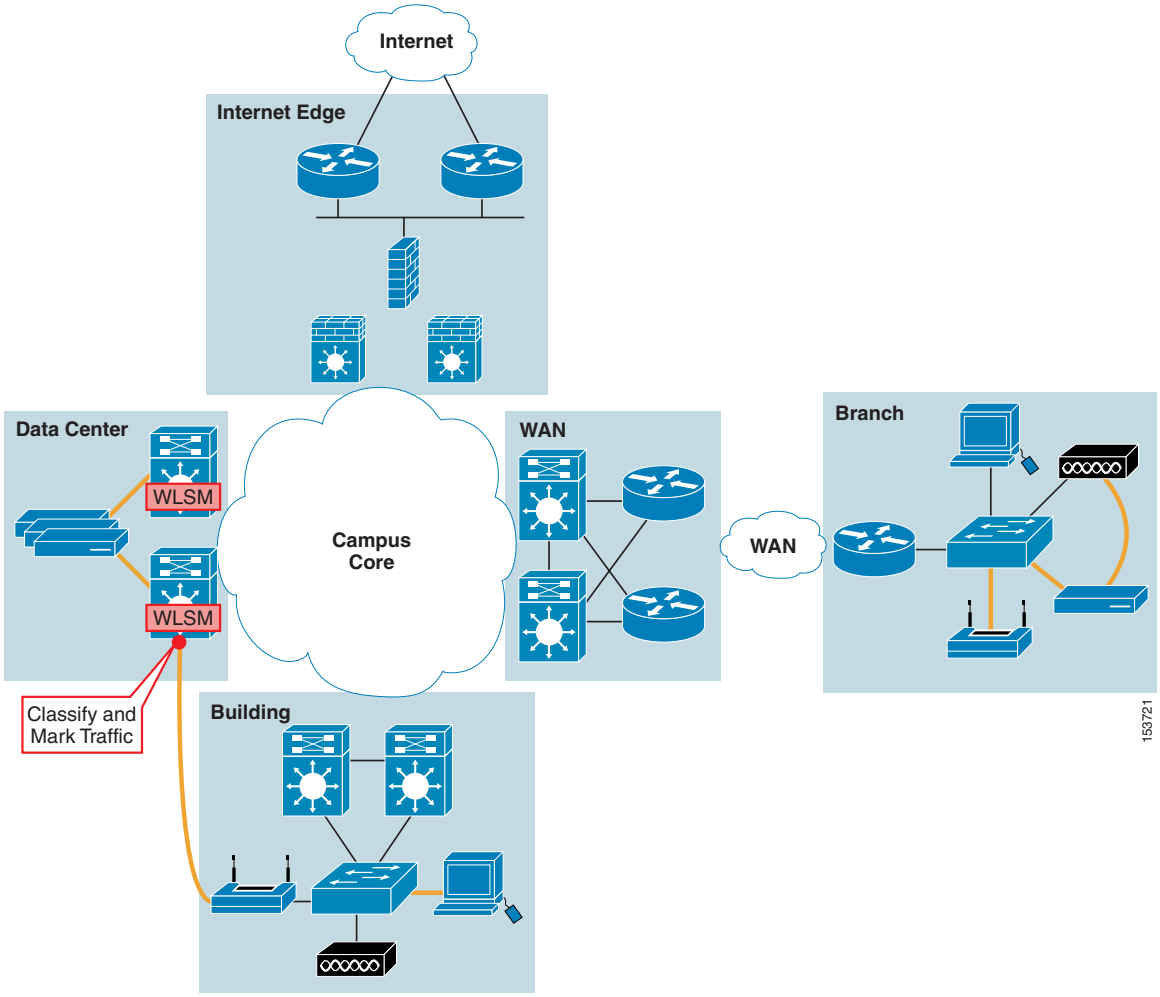
```
3560-access(config)#policy-map EDGE-GRE-POLICY
3560-access(config-pmap)#class EDGE-GRE
3560-access(config-pmap-c)#police 1000000 8000 exceed-action drop
    "Apply the policer on the switch interfaces
```

```
3560-access(config)#interface FastEthernet0/34
3560-access(config-if)#service-policy input EDGE-GRE-POLICY
```

Dynamic Approach

Marking of the decapsulated traffic at the centralized location is the recommended choice when deploying a dynamic approach. This is done on a Catalyst 6500 equipped with Sup720; UBRL is the logical choice. The policer can be applied on the mGRE interface receiving all the edge traffic, to apply the marking before sending it into the core, as shown in Figure 27.

Figure 27 Policing Applied on the mGRE Interface at the Central Location



The required configuration steps are as follows.

Step 1 Define the class map to identify the edge traffic:

```
6500-DC(config)#access-list 101 permit ip any any
6500-DC(config)#class-map match-all EDGE
6500-DC(config-cmap)#match access-group 101
```

Step 2 Define the policer to be applied on the mGRE interface. Mark all traffic that exceeds the specified threshold (1 Mbps in the example) as scavenger traffic (not dropped).

```
6500-DC(config)#mls qos map policed-dscp normal 0 to 8
6500-DC(config)#policy-map EDGE-POLICING
6500-DC(config-pmap)#class EDGE
6500-DC(config-pmap-c)#police flow mask src-only 1000000 8000 conform-action
set-dscp-transmit 0 exceed-action policed-dscp-transmit
```

Step 3 Apply the policer:

```
6500-DC(config)#interface Tunnel 10
6500-DC(config-if-range)#service-policy input EDGE-POLICING
```

When applying an inbound policy map on the mGRE logical interface, the same considerations proposed in Catalyst 6500 with Cisco IOS, page 57 are still valid. If the traffic is eventually GRE-encapsulated before being sent out, only the outer IP header has the DSCP field marked correctly.

Because the GRE traffic originated on the distributed access points, it must be sent across the campus core to get aggregated on the Catalyst 6500 equipped with WLSM. Optionally, you can mark it on the access layer device where the access point is connected.

As mentioned in the section covering the static approach, you cannot use the per-port/per-VLAN functionality, so you must configure an ACL matching the destination address of the GRE tunnel originated on the access point and associate it to the user SSID. Following is a sample configuration that is valid for a Catalyst 3560, and easily extendable to other Catalyst platforms:

Step 4 Define the class map to identify the edge traffic:

```
3560-access(config)#access-list 110 permit gre any host 10.121.253.254
3560-access(config)#class-map match-all EDGE-GRE
3560-access(config-cmap)#match access-group 110
```

Step 5 Define the policer. Mark all traffic that exceeds the specified threshold (1 Mbps in the example) as scavenger traffic (not dropped):

```
3560-access(config)#mls qos map policed-dscp normal 0 to 8
3560-access(config)#policy-map EDGE-GRE-POLICY
3560-access(config-pmap)#class EDGE-GRE
3560-access(config-pmap-c)#set dscp default
3560-access(config-pmap-c)#police 1000000 8000 exceed-action policed-dscp-transmit
```

Step 6 Apply the policer on the switch interfaces:

```
3560-access(config)#interface FastEthernet0/34
3560-access(config-if)#service-policy input EDGE-GRE-POLICY
```

WLAN Controller

Deploying WLAN controllers in the campus network implies that all traffic is tunneled from the edge access points to the controllers that can be deployed, for example, in a centralized location such as the campus data center. This behavior is very similar to the WLSM-based scenario described previously. The main differences are that traffic is tunneled using Lightweight Access Point Protocol (LWAPP) (and not GRE), and that the configuration of all the access points is performed centrally from the controller.

Static Approach

Differently from WLSM deployments, in this case keep in mind that the same LWAPP tunnel is used to carry data traffic for users belonging to different groups (usually associated using different SSIDs). As a result, it is not possible to classify the traffic for a specific user group on the access layer switch where the access point is connected. The only option is then to classify and rate limit it when it is bridged on

the corresponding VLAN at the WLAN controller location. The platform where this is accomplished can vary, but is most likely a Catalyst 6500 when deploying the WLAN controllers in a centralized location (such as a data center), or when using the WLSM.

Following is a sample configuration that is valid for a Catalyst 6500:

Step 1 Define the class map to identify the edge traffic:

```
6500-DC(config)#access-list 101 permit ip any 10.124.150.0 0.0.0.255
6500-DC(config)#class-map match-all EDGE-TRAFFIC
6500-DC(config-cmap)#match access-group 101
```

Step 2 Define the policer:

```
6500-DC(config)#policy-map EDGE-TRAFFIC-POLICING
6500-DC(config-pmap)#class EDGE-TRAFFIC
6500-DC(config-pmap-c)#police flow mask dest-only 1000000 8000 conform-action
set-dscp-transmit 0 exceed-action drop
```

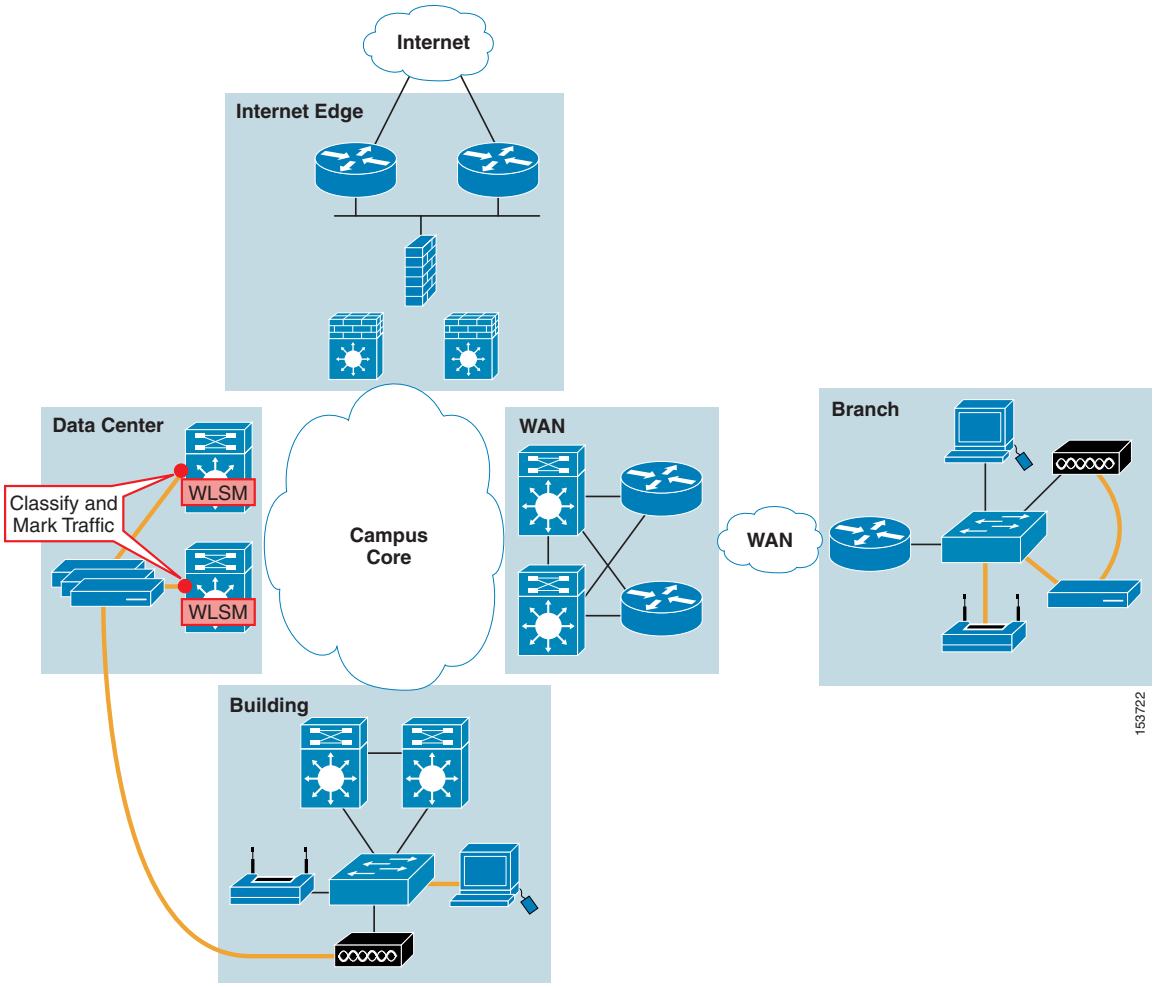
Step 3 Apply the policer on the switch VLAN interface:

```
6500-DC(config)#interface Vlan 150
6500-DC(config-if-range)#service-policy input EDGE-TRAFFIC-POLICING
```

Dynamic Approach

Once again, the dynamic approach consists in marking out-of-profile traffic as scavenger traffic. Following decapsulation, the traffic is bridged to a unique VLAN that is associated to the WLAN, so Cisco recommends that you mark the traffic on the switch to which the controller is connected, as shown in Figure 28.

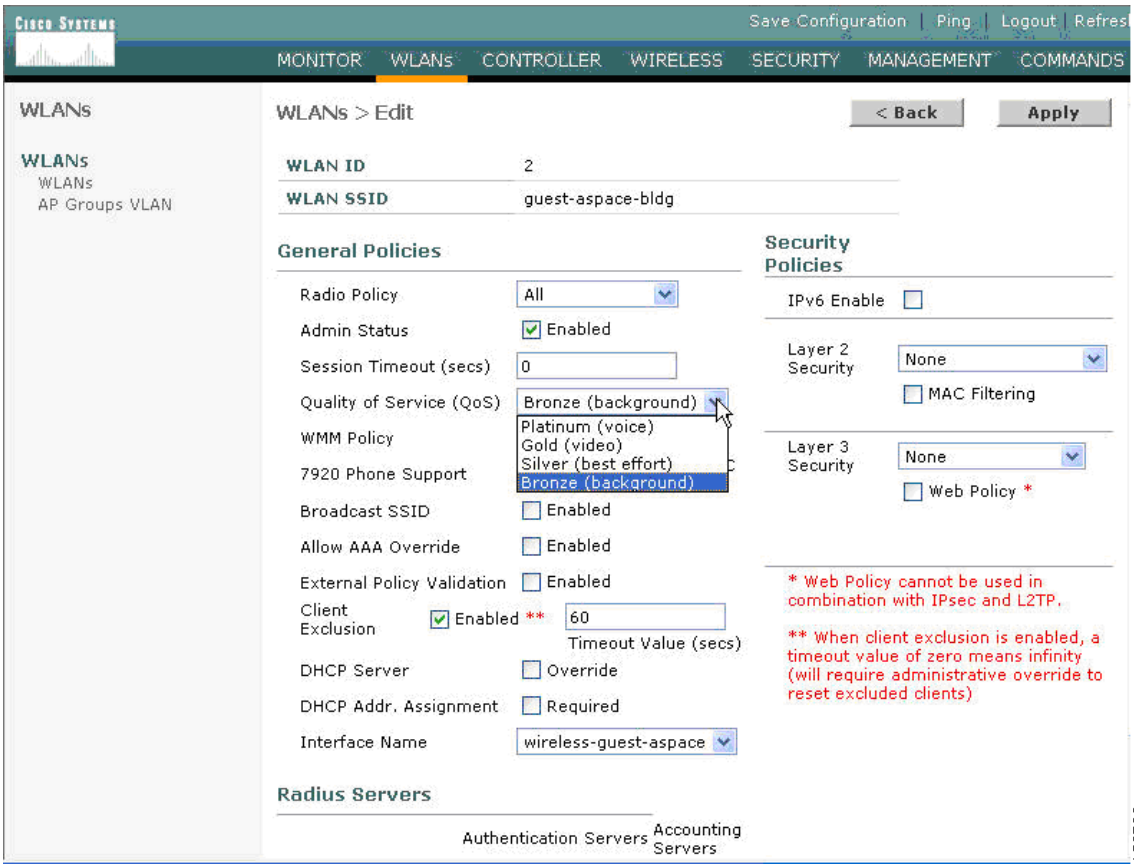
Figure 28 Marking LWAPP-Decapsulated Traffic



Depending on the specific platform to which the controller is connected, you can perform the same type of marking strategy that is described in Figure 28.

Additionally, when a WLAN is created on the controller, it is possible to associate a QoS level to it (see Figure 29).

Figure 29 Selecting a QoS Level for a WLAN



Depending on the level selected, the access point marks the DSCP for upstream traffic. The DSCP is set in the external IP header (traffic is LWAPP-encapsulated), as shown in Table 3.

Table 3 Default DSCP Marking on LWAPP APs

Class	DSCP
Platinum	46
Gold	26
Silver	0
Bronze	10

With the adoption of the Cisco QoS Baseline (starting in 2002), Cisco does not recommend using terms such as platinum, gold, silver, and bronze to describe QoS classes, because such terms do not accurately convey the service level requirements of the applications within the classes. Furthermore, such terms seem to convey an oversimplified and often inaccurate strict application hierarchy. The following is per the QoS baseline:

- DSCP 46 is the default marking for a voice class.
- DSCP 26 (also referred to as AF31, as defined in RFC 2597) is the default marking value for a locally defined mission-critical data class.
- DSCP 0 is the default marking for the best effort class (per RFC 2474).

- DSCP 10 (also referred to as AF11, as defined in RFC 2597) is the default marking value for a bulk data class.

As shown in Table 3, the bronze setting does not correspond to a scavenger value (CS1 or 8), but to bulk (10). As a result, there are the following two options:

- Leave the default marking for LWAPP-encapsulated traffic and configure the queuing strategy on all the devices between the access point and the controller so that this type of traffic is handled in a similar manner as the scavenger class.
- Mark the LWAPP-encapsulated traffic on the first access layer switch where the access point is connected, similar to what is suggested for GRE traffic in the WLSM scenario. If using this approach, the selection of the QoS level for the WLAN becomes meaningless because the traffic is marked anyway.

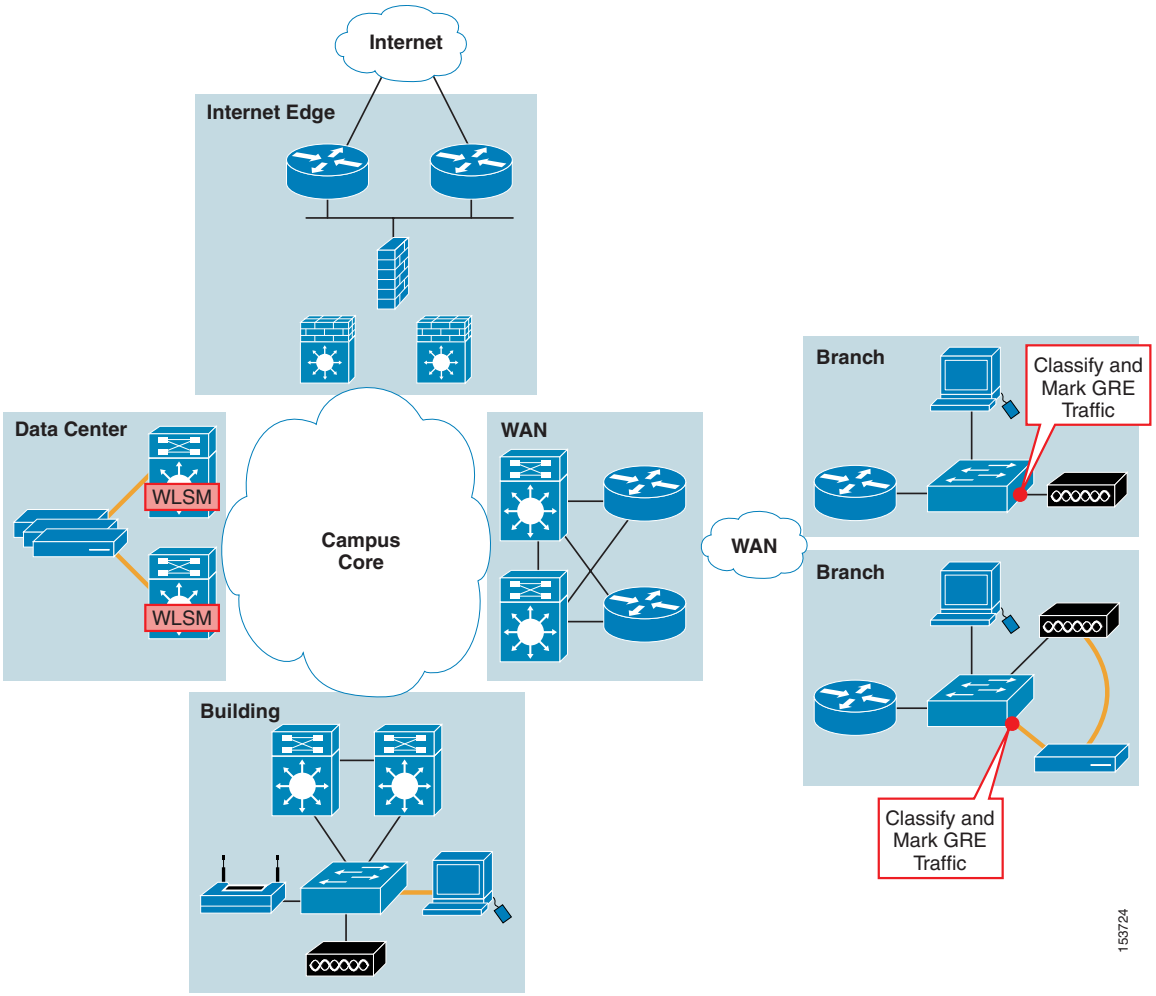
Note As described in WLSM , page 60, the marking of traffic is optional. Cisco recommends marking the LWAPP-decapsulated traffic that is bridged by the WLAN controller on the corresponding VLAN. This should be always done, considering that any previous marking that applied to LWAPP traffic is lost when the traffic is decapsulated on the controller.

For branch deployments, there are the following two options when deploying WLAN controllers:

- In the first option, a local WLAN controller is deployed at the branch location. In this case, the same considerations given for campus deployments can be followed.
- In the second option, remote edge access points are deployed at the branch location to locally bridge the user traffic. In this case, the classification and marking of traffic can be accomplished in the same manner as the wired case.

Both these options are shown in Figure 30.

Figure 30 Classifying and Marking Traffic at the Branch for WLAN Controller Deployments



Challenges and Limitations Using VRF and GRE

As described in previous sections, it is clear that the use of VRF and GRE to build VPNs inside the campus network provides many advantages when compared with the distributed ACLs approach. These advantages include the support of overlapping address spaces between VPNs, the path differentiation capabilities offered by the use of a separate routing table per VPN, and the perception of the achievement of a safer solution.

However, the VRF and GRE solution should be implemented only in applications for which it is well-suited, because of the following limitations:

- Configuration intensity—As previously mentioned, building a VPN using VRF and GRE is well-suited for applications required hub-and-spoke connectivity. In scenarios where any-to-any connectivity must be achieved, the configuration task in building GRE tunnels connecting all the various sites of the network can quickly become unmanageable. The use of mGRE helps in simplifying the configuration, but it is minimized by the limited level of support on platforms normally deployed in campus networks.

- Limited scalability and performances—As discussed in Connectivity Requirements, page 21, GRE is supported in hardware only on Catalyst 6500 switches equipped with Supervisor 32 or 720. As a result, the scalability and performance that can be achieved with this solution are tightly linked to the specific devices deployed in the network. Also, for designs where deployed platforms supporting GRE in software (such as Catalyst 4500 switches), additional precautions must be taken to protect the CPU of these devices from becoming over-used. The recommended way to achieve this is by rate limiting the traffic.

Path Isolation Deploying MPLS VPN

Multiprotocol Label Switching (MPLS) has traditionally been viewed as a service provider (SP) routing technology: SPs have commonly used MPLS VPN to create tunnels across their backbone networks for multiple customers. In that way, individual customer traffic is carried on a common service provider network infrastructure. Using the same principle, MPLS VPN can be deployed inside the enterprise network to logically isolate traffic between users belonging to separate groups (as for example guest, contractors, and employees) and to provide a technical answer to the business problems discussed at the beginning of this guide.

The main advantage of MPLS VPN when compared to the other path isolation technologies previously discussed is the capability of dynamically providing any-to-any connectivity without facing the challenges of managing many point-to-point connections (as for example is the case when using GRE tunnels). MPLS VPN facilitates full mesh of connectivity inside each provided segment (or logical partition) with the speed of provisioning and scalability found in no other protocol. In this way, MPLS VPN allows the consolidation of separate logical partitions into a common network infrastructure.

The following sections of this guide describe the steps required to enable MPLS VPN end-to-end across the enterprise network. The initial section presents a quick overview of the MPLS VPN technology; the assumption here is that the reader is already familiar with the technology, so the purpose of this specific section is simply to review how the technology works and what are the various technical components involved. After that, the focus shifts to deploying MPLS VPN in an enterprise campus environment: the goal here is to provide design guidance for applying MPLS VPN to the enterprise campus and analyze the impact that has on a campus network configured following the recommended and consolidated design. The design considerations are provided based on some initial assumptions that are discussed in Path Isolation Initial Design Considerations, page 12, and that are reviewed in this section. Finally, various alternatives to extend logical isolation across the WAN to extend the VPNs to remote branch locations are discussed.

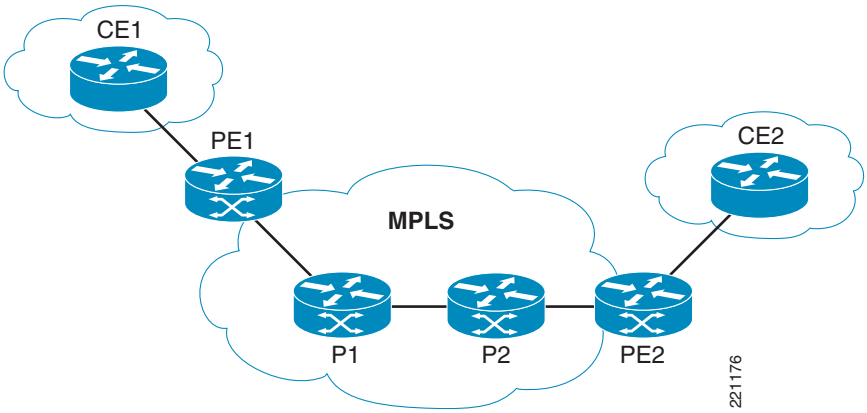
MPLS VPN Technology Overview

MPLS Rehearsal

As already mentioned in the previous section, MPLS was originally deployed for the service provider environment. This heritage becomes more evident when describing the various roles that the network devices perform in an MPLS-enabled network.

Figure 31 shows the three roles a device can play when deploying MPLS.

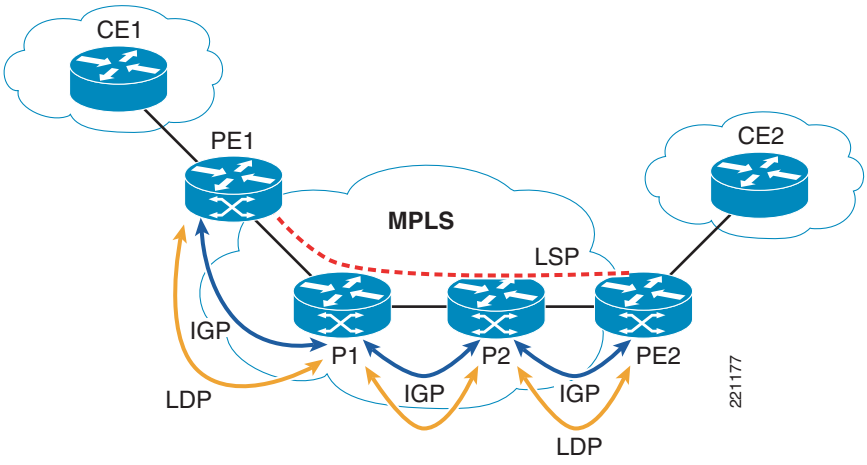
Figure 31 Device Roles in an MPLS Network



1. Customer edge (CE) router—This is traditionally the network device at the customer location that interfaces with the service provider. In Figure 31, CE1 and CE2 represent the routers at the customer remote locations that need to be interconnected via the MPLS service provider network.
2. Provider edge (PE) router—This is the device at the edge of the service provider network that interfaces with the customer devices. The PE devices are often also called label switching routers edge (LSR-Edge), because they sit at the edge of the MPLS-enabled network.
3. Provider (P) router—These are the devices building the core of the MPLS-enabled network. Their main functionality is to label switch traffic based on the most external MPLS tag imposed to each packet and for this reason are often referred to as label switching routers (LSRs)

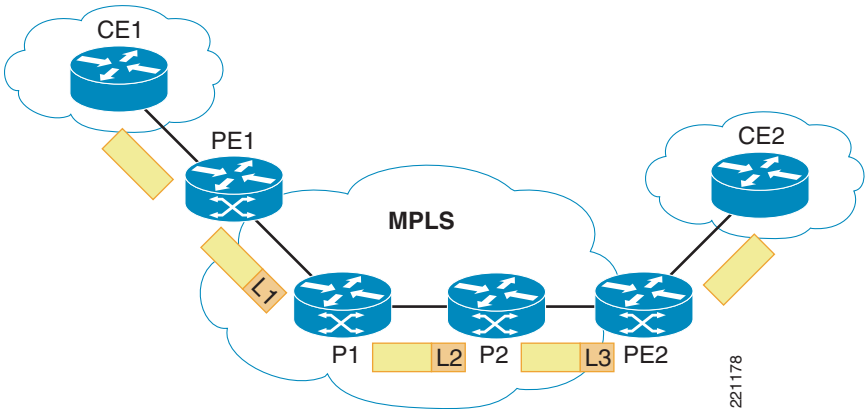
From a control plane point of view, an MPLS-enabled network uses two separate protocols: first, an IGP running in the core of the network and providing connectivity between the various network devices. Second, a Label Distribution Protocol (LDP) providing a standard dynamic methodology for hop-by-hop label distribution in the MPLS network. LDP works by assigning labels to routes that have been chosen by the underlying IGP routing protocol. The resulting labelled paths, shown in Figure 32 and called label switched paths (LSPs), forward label traffic across an MPLS backbone to particular destinations.

Figure 32 MPLS Control Plane



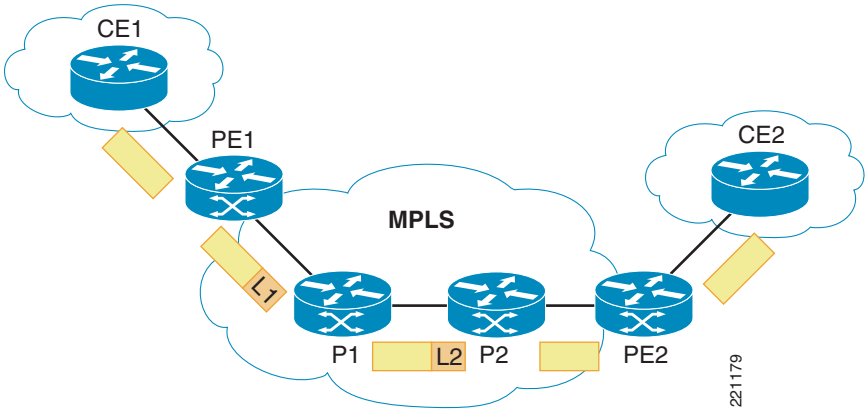
From the point of view of data forwarding, traffic that needs to be sent between remote customer sites is label-switched along the LSP, as shown in Figure 33.

Figure 33 MPLS Data Plane



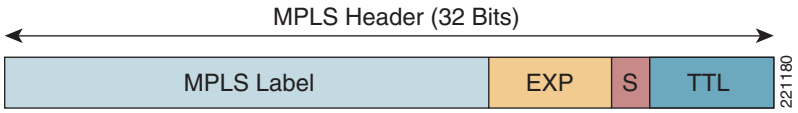
Each device along the LSP switches the traffic based on the incoming MPLS label; a new tag is imposed before the packet is sent to the next device. Notice that the behavior shown in Figure 33 may be in reality slightly different because of a functionality called Penultimate Hop Popping (PHP). By default the egress PE device explicitly informs the neighbor P not to tag packets directed to it, so that the PE can switch the packet based only on IP information without having to do a double lookup (first one for the MPLS tag, second one for the IP information). Figure 34 shown the same network above when using PHP.

Figure 34 Penultimate Hop Popping



The MPLS tag shown in Figure 34 is a 32 bit header that is structured as shown in Figure 35.

Figure 35 MPLS Label



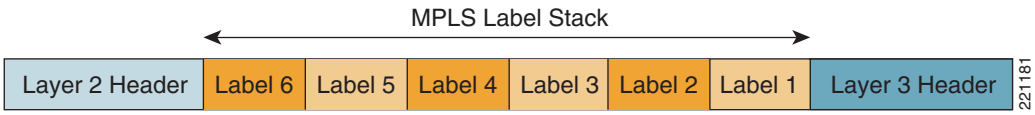
The structure is as follows:

- MPLS Label—20-bit field used for label switching the packet and is replaced at every hop in the MPLS network

- EXP—3-bit field that is used to indicate the class of service (CoS) of the MPLS packet (similarly to the CoS field in Ethernet frames)
- S—Bit used to indicate the bottom of the stack when more than one MPLS label is imposed on the packet (as seen subsequently in the case in the MPLS VPN scenario)
- TTL—8-bit time-to-live value (having the same functions of loop detections as the homonymous IP field)

The MPLS label is placed after the L2 headers for a packet. Notice that a packet can have multiple MPLS labels appended to it; this is referred to as the label stack. Each MPLS label has a specific meaning for the node that pushed the label onto the packet, and the node that pops that label from the stack. The LSR routers in the network forward packets only based on the outer most label. The lower labels are taken into account only when they become the outermost label after the previous outermost label has been popped. MPLS labels are pushed onto packets starting with the original frame, and additional labels are added on top of the outer most label. MPLS labels are popped starting with the outer most label, the last one pushed onto the label stack. (See Figure 36.)

Figure 36 MPLS Label Stack

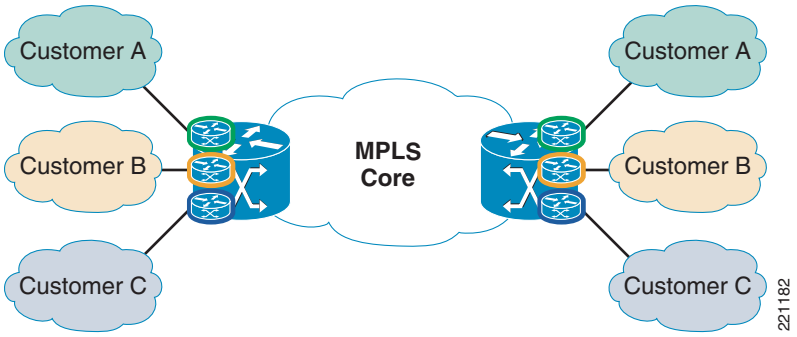


Another important concept widely used when discussing MPLS is the forwarding equivalence class (FEC). An FEC is a set of packets that all meet some defined criteria, and are forwarded in the same way by a router. The packets can differ from each other from the information carried in the network layer (source, destination addresses, and ToS) but are forwarded using the same rule. An example of an FEC is all unicast packets destined to a particular prefix. They can have different destination addresses but the destination addresses all fall under the same prefix. The forwarding entry that a router maintains for a packet contains the classification criteria (normally destination address) and the next hop address. Packets that fall into an FEC associated with a particular forwarding entry are forwarded to the next hop router specified by the entry. Note that an FEC in the world of IPv4 routing is nothing more than a prefix in the routing database; this essentially implies that a separate LSP is built for each individual routing database entry.

MPLS VPN Rehearsal

The discussion above applies to a scenario where the MPLS network is used to connect remote sites belonging to the same customer organization. For the SP to use the same MPLS core to provide connectivity services to different customers, as shown in Figure 37, something more than MPLS is needed, which is MPLS VPN.

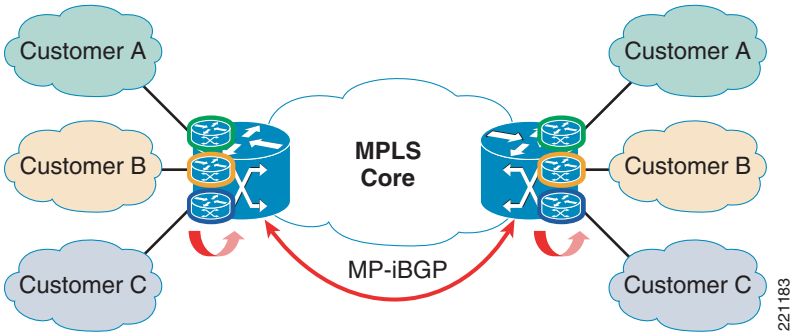
Figure 37 MPLS VPN



The key technology that simplifies the deployment of MPLS VPN is VRF, which is discussed in Control Plane-Based Path Isolation, page 8. As shown in Figure 37, defining distinct VRF instances on each PE device allows separating the traffic belonging to different customers, allowing for logical isolation and independent transport across the common MPLS core of the network. Notice that the VRF definition is required only on the PE devices, whereas the P routers in the core of the network have no knowledge of VRFs; they simply label-switch traffic based on the most external MPLS label.

From a control plane perspective, an additional component now needs to be added to the IGP and LDP protocols previously discussed: Multi-Protocol BGP (MP-BGP), which is used as the mechanism to exchange VPN routes between PE devices. As shown in Figure 38, for this to work, an MP-iBGP session needs to be established between all the PE devices (in a fully meshed fashion).

Figure 38 Control Plane for MPLS VPN

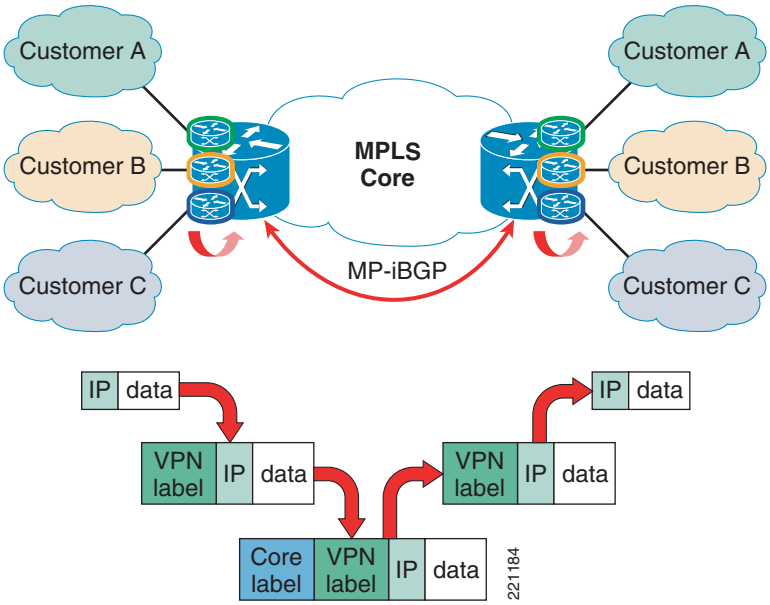


From a control plane perspective, the following two important elements need to be defined to perform the exchange of VPN routes through MP-BGP:

- Route distinguisher (RD)—Represents a 64-bit field (unique for each defined VRF) added to each 32-bit IPv4 address to come up with a unique 96-bit VPN IPv4 prefix. This ensures the uniqueness of address prefixes across different VPNs, allowing support for overlapping IPv4 addresses.
- Route target—Represents an extended attribute exchanged through MP-BGP and allows the PE devices to know which routes need to be inserted into which VRF. Every VPN route is tagged with one or more route targets when it is exported from a VRF (to be offered to other VRFs). It is also possible to associate a set of route targets with a VRF, so that all the routes tagged with at least one of those route targets are inserted into the VRF.

From a data plane perspective, the packets belonging to each VPN are labeled with two tags: the internal tag uniquely identifies the specific VPN the packets belong to, whereas the external tag is used to label-switch the traffic along the LSP connecting the ingress PE toward the egress PE. This concept is highlighted in Figure 39.

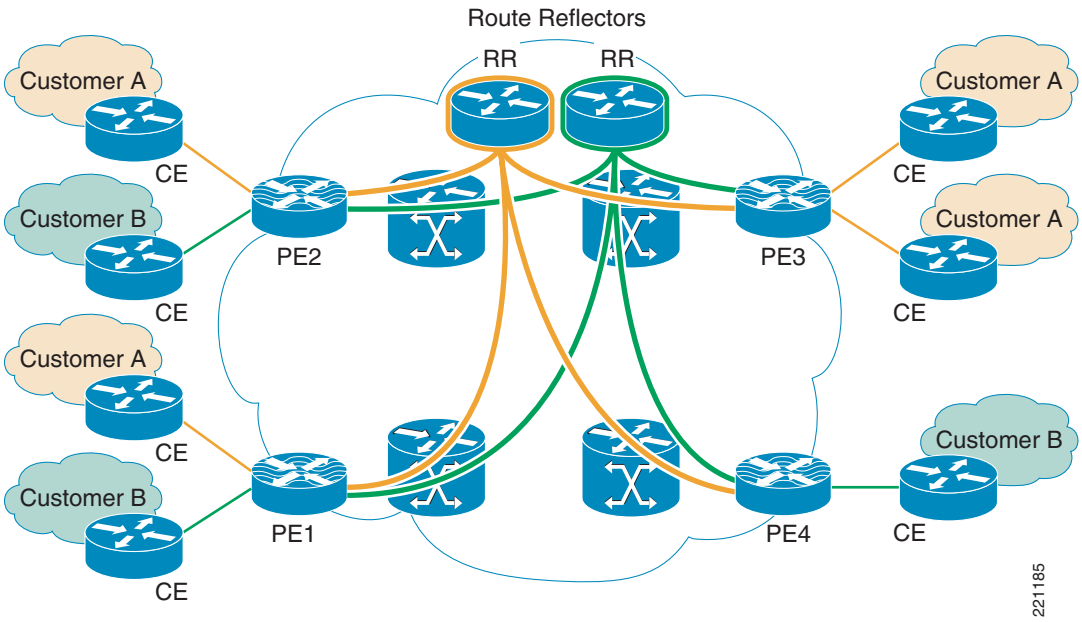
Figure 39 Data Plane for MPLS VPN



As shown in Figure 39, when the IP packet is received at the ingress PE, a first VPN label is imposed on it. The information on what VPN label to apply has been received from the egress PE via MP-iBGP. Before sending the packet to the MPLS core, the ingress PE must also impose a second tag (the most external one), which is used to label switch the packet along the LSP connecting the ingress PE to the egress PE. When the egress PE receives the packet, it is able to look at the VPN label and based on that specific label, send the traffic in the proper VPN.

Finally, the last element that needs to be considered for an MPLS VPN deployment is the route reflector (RR). Because MP-iBGP sessions need to be established between the different PEs defined at the edge of the MPLS network, Cisco usually recommends not deploying a full mesh of iBGP connections but instead using several devices as route reflector routers.

Figure 40 Deployment of Route Reflectors



Each route reflector peers with every PE device (in a hub-and-spoke fashion), contributing to the overall stability of the design. Also, deploying route reflectors eases the addition of new sites, because only a new peering with the route reflector needs to be established without modifying the configuration of the remaining PE devices. The following paragraph highlights the advantages of deploying route reflectors both in a campus and WAN environments.

MPLS VPN in Campus

High Level Design Principles

Current campus networks must address a new set of customer requirements, such as the desire for mobility, the drive for heightened security, and the need to accurately identify and segment users, devices, and networks. All these drivers are leading enterprises to revisit their campus design requirements.

The Cisco-recommended design for the campus network is architected in a hierarchical model comprised of core, distribution, and access that provide distinct features and functionalities. Multilayer designs using Layer 2 in distribution and access enable the design of modular topologies using scalable “building blocks” that allow the network to meet evolving business needs. The multilayer model based on modular design is easy to scale, understand, and troubleshoot because it follows a deterministic traffic pattern.

An in-depth discussion of Cisco-recommended campus network design is out of the scope of this guide. For more information on this topic, see the following URL:
http://www.cisco.com/en/US/netsol/ns656/networking_solutions_design_guidances_list.html#anchor2

When deploying MPLS VPN in a campus environment, keep in mind the following two key points:

- The assumption is that the campus network should be always deployed following the recommended design principles highlighted in the documents referenced above.

- Understanding what modifications (or simplifications) need to be applied to an SP-based technology to fit within the enterprise, while trying to maintain the campus MPLS deployments as simple and straightforward as possible. This means that deploying network virtualization should not impact "what is already working" in the network. In addition, even inside each logical partition, the user should experience the same characteristics of scalability, hierarchy, stability, and so on, as if the user was part of a dedicated physical infrastructure.

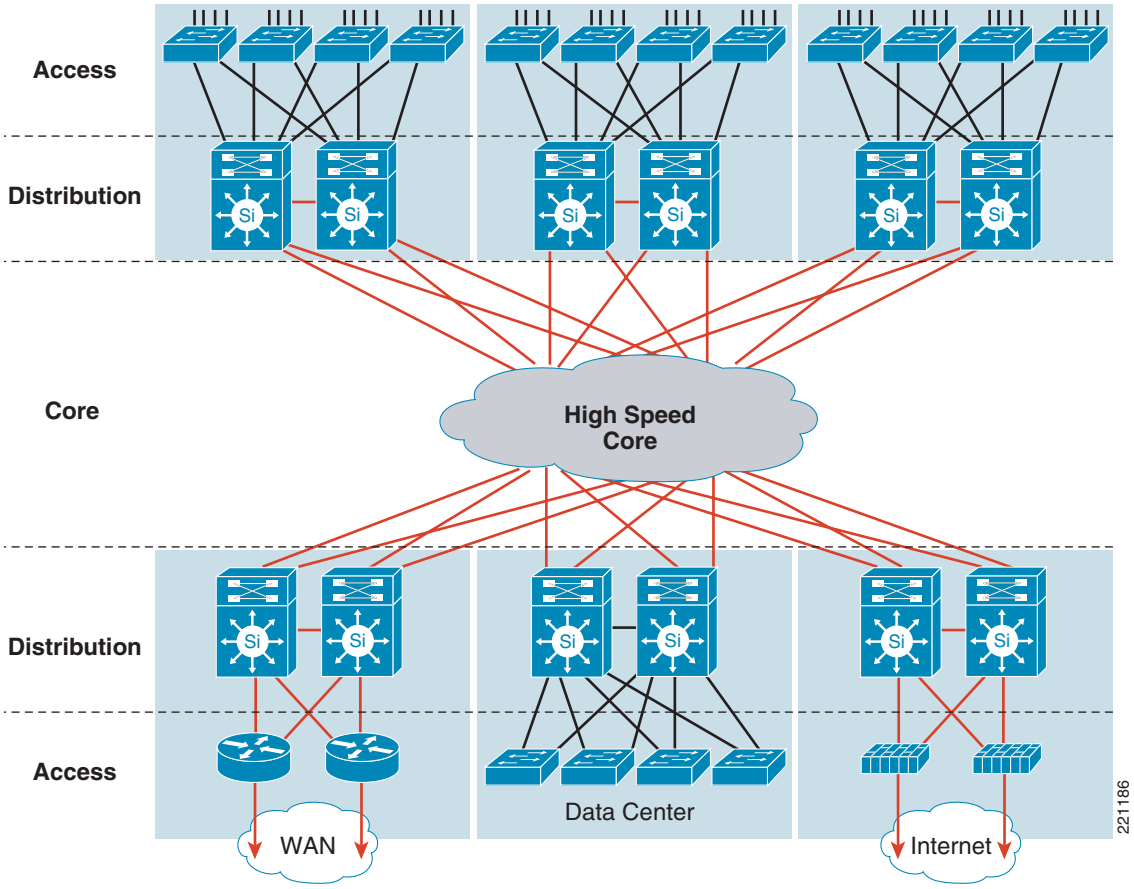
The general considerations made in Path Isolation Initial Design Considerations, page 12 are also valid also when deploying MPLS VPN as a path isolation option; it is thus recommended to read that specific section to properly frame the solution. In addition to that, some important design principles or differences between an enterprise MPLS-VPN and an SP deployment need to be kept in mind when specifically deploying MPLS VPN in a campus environment. The following assumptions are also uniquely characterizing these deployments from the traditional service provider ones.

- New design principles related to the IGP deployment now need to be kept in mind:
  - The IGP used in the global table runs edge-to-edge across the enterprise network, differently from an SP-like MPLS VPN deployment, where usually it is confined in the core.
  - There are no longer customer IGPs running at the edge of the network whose routes are tunneled across the backbone.
  - The IGP in use is providing multiple functions. On one side, it allows to establish MP-iBGP sessions between the PE devices deployed at the edge of the MPLS domain and to setup the LDP sessions required for exchanging MPLS labels between neighbor devices. At the same time, it is also used to allow network connectivity to the entities that remain in the global table. This is very relevant when deploying Virtual Networks for specific purposes (Guest/Partner Access, NAC Remediation, and so on), because it is expected that most of the internal enterprise traffic still remains switched in the global table.
- The IGP used in the global table has a double functionality: on one side, it allows the establishing of MP-iBGP sessions between the PE devices deployed at the edge of the MPLS domain and to exchange MPLS labels through a specific LDP protocol. At the same time, it is also used to allow network connectivity to the entities that remain in the global table. As already mentioned, the current recommendation is to use virtual networks only for specific purposes. This means that most of the internal enterprise traffic still remains switched in the global table. This represents a first differentiation from the SP-like MPLS VPN deployment, because in that case the global table is usually used to provide only PE-PE connectivity and does not extend to the edge of the network but only remains in the core.
- The solution discussed here constitutes an evolutionary or overlay design. The goal of this design is to use MPLS VPN to provide additional services within an existing network to complement rather than replace the existing campus network.
- The MP-BGP process represents the control plane that allows the establishment of forwarding paths for VPN traffic and is used in addition to the IGP that perform the same functionality for IPv4 global traffic. As a consequence, a single AS scenario is discussed in this phase of the project: this implies that the routing protocol in global table (IGP) extends end-to-end in the enterprise network (campuses, data centers, and remote offices). MP-BGP is thus overlaid on top of the IGP running in the global table.
- Enterprise design requires end-to-end operational support processes. The division between PE and CE devices exists now technically but not operationally, because both are now part of the same enterprise network and as such are most likely administered by the same group. Also, it is worth noting that in many cases, there is no CE device or role in the design either, because when deploying MPLS VPN in multilayer campus networks, all the edge VPN subnets results directly connected to the PE devices.

Network Topologies

One of the main goals of this guide is to determine the impact of turning on MPLS VPN in a working campus network environment deployed based on the hierarchical design recommendations. Figure 41 shows an example of a hierarchical campus network. The various campus distribution blocks are connected by a high speed core. The assumption here is that these connections are point-to-point links and that the enterprise has control of all the devices building the high speed core.

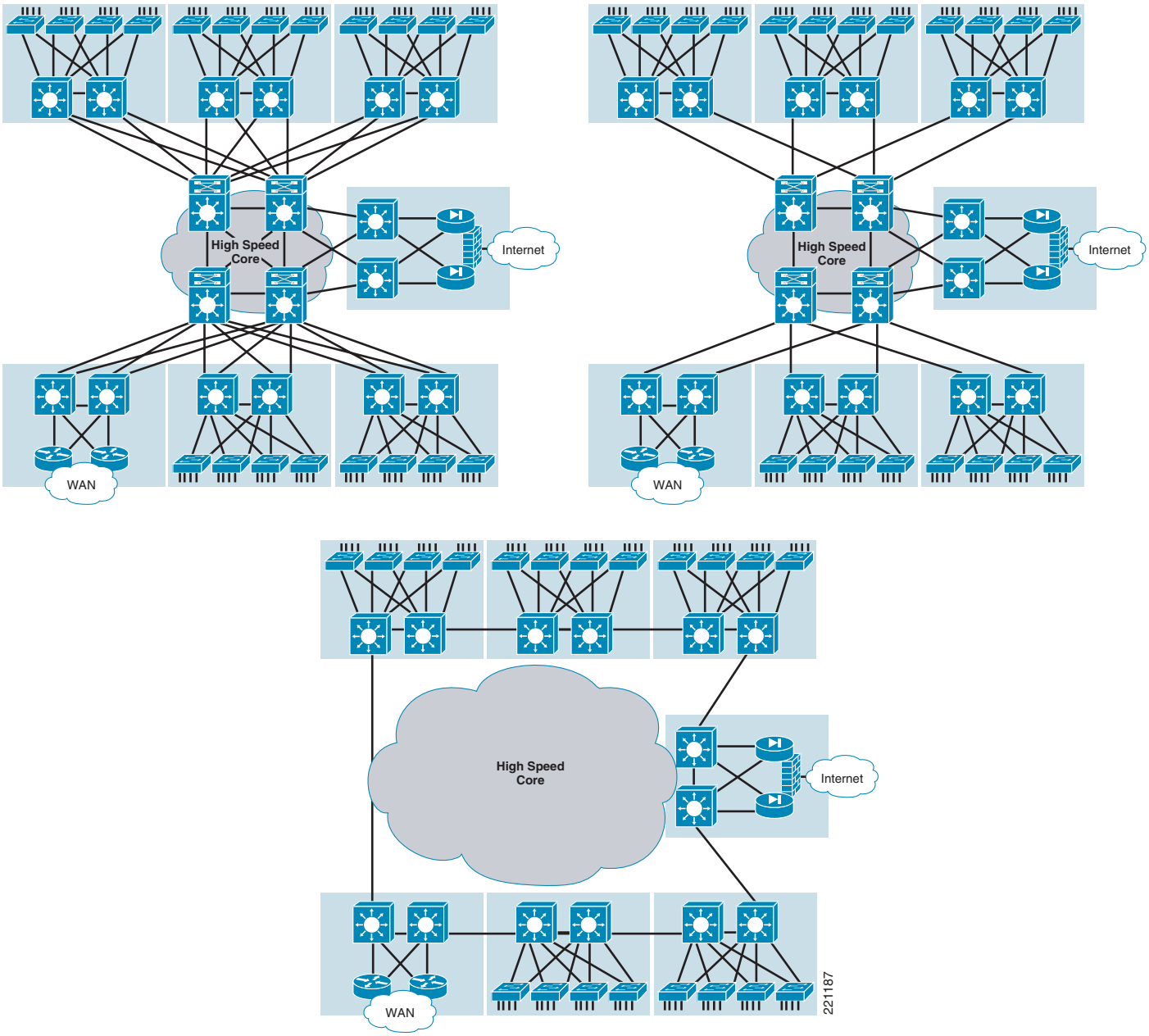
Figure 41 Hierarchical Campus Network



Starting from the general campus model shown in Figure 41, three main topologies are analyzed in the following sections, as represented in Figure 42:

- Fully-meshed topologies
- Partially-meshed topologies
- Ring topologies

Figure 42 Campus Network Topologies



These represent three common topologies that are often deployed. Although it is always recommended that the core network design implement a full mesh topology whenever possible, it is relevant to note that there is often not the possibility of connecting the core devices in a fully-meshed fashion because of cost or geographical location issues. In such hybrid scenarios, each building block would be fully meshed to the core devices, and the core devices would be linked in a ring fashion. The campus fully-meshed design is traditionally the recommended one for its characteristics of convergence, reliability, and traffic load balancing. This recommendation holds true when deploying MPLS VPN in the campus environment. However, because following this guideline may not always be possible in real network deployments, the following sections highlight possible issues to keep in mind when deviating from the ideal fully-meshed scenario.

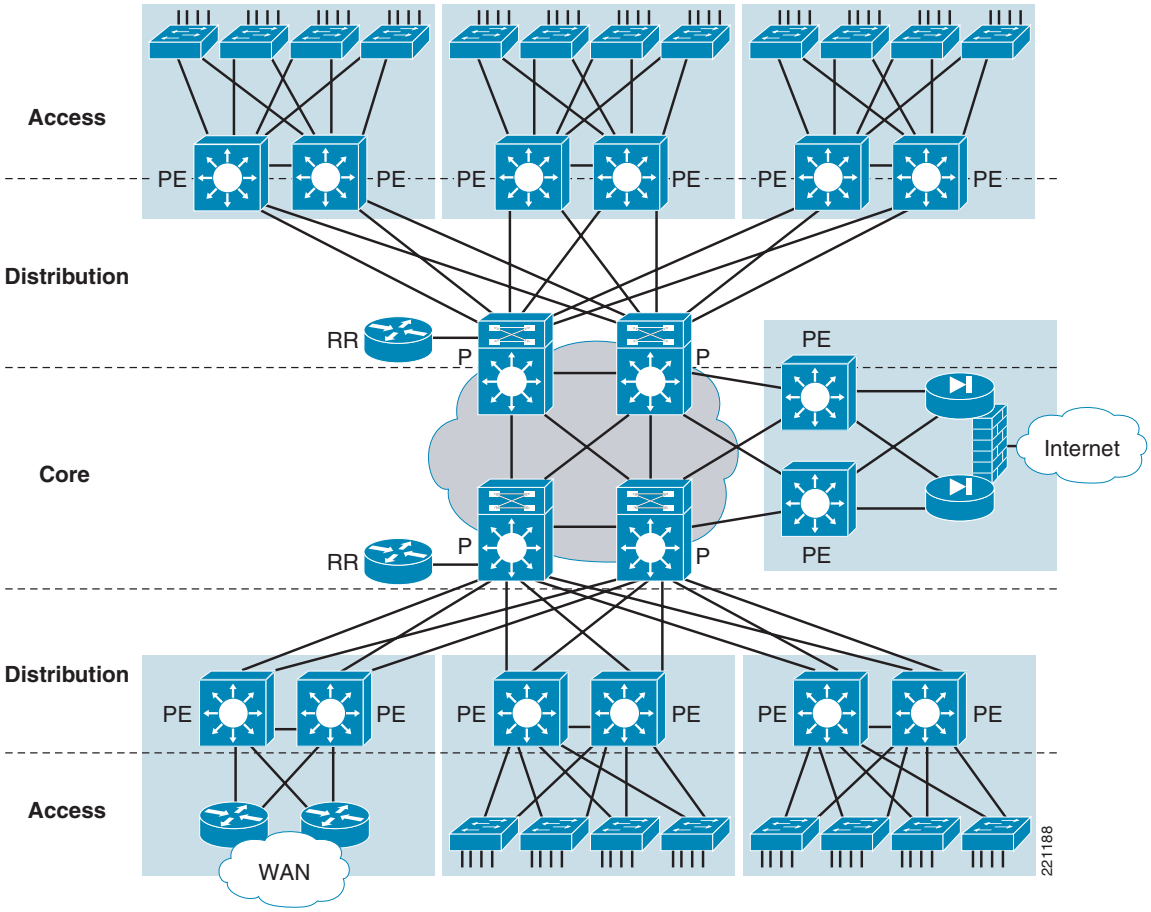
Network Device Roles

As discussed in MPLS VPN Technology Overview, page 69, when deploying MPLS VPN, there are essentially four roles that the device can play in the design: CE, PE, P, and route reflectors (RRs).

In a traditional multilayer campus design, the access layer devices are L2 capable and the first L3 hop in the network is at the distribution layer. Core nodes are L3 routed devices interconnecting various campus distribution blocks.

When deploying MPLS VPN as an overlay model in such campus environment, the recommended roles and positioning for the network devices involved in the deployment are shown in Figure 43.

Figure 43 Device Roles in an MPLS Network



As shown in Figure 43, the PE devices are positioned at the first L3 hop in the network, which is the distribution layer. VRFs must in fact be defined at the first L3 hop device, to extend at L3 the logical isolation provided by VLANs at L2. As a consequence, the recommendation is to deploy there a platform supporting VRF capabilities and capable of performing MPLS label-switching functionalities.

Note In designs where the platforms deployed at the distribution layer are not MPLS capable, the use of some other technique (such as VRF-lite) is required to extend the VRF isolation to a PE device deployed in the core. Discussing this model is out of the scope of this guide, so the assumption here is that MPLS-capable devices are deployed in the distribution layer of the campus network.

Deploying PE functionalities at the distribution layer implies that all the other devices constituting the high speed core of the network play the P role. Note how in the specific design shown in Figure 43, there are actually no true CE devices, because the only entities connecting to the PE (except for the P switches) are access layer switches that perform only L2 functionalities. Finally, Cisco recommends using two additional routers as RRs, connecting them to the core devices.

Note RR deployment is further discussed in MP-iBGP Deployment Considerations, page 108.

VRF and MPLS on Catalyst 6500 Platforms

The only switching platform commonly deployed in campus networks currently supporting MPLS is the Catalyst 6500 equipped with Sup720 or Sup32 PFC3B or DFC3B (and higher). Having an understanding of the operation of label switching on this device helps in comprehending the design and how to better troubleshoot eventual issues discussed subsequently in MPLS-Specific Troubleshooting Tools, page 139.

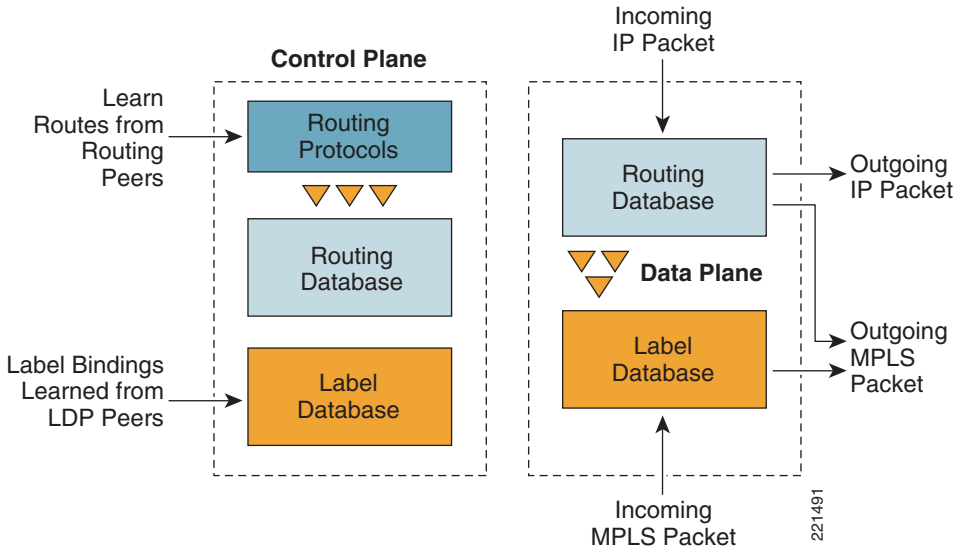
Note MPLS is supported only on 6500 platforms running Cisco IOS (Native) and not in a Hybrid (CatOS + IOS) system.

For a basic understanding of packet forwarding in the Catalyst 6500 architecture and for more information on terms such as PFC, DFC, and CEF, see the following URL: http://www.cisco.com/en/US/partner/products/hw/switches/ps708/index.html

Hardware Components involved in MPLS Switching

To understand the various platform components involved in MPLS switching, it is necessary to distinguish between control and data planes, as shown in Figure 44.

Figure 44 High Level View of Control and Data Planes on Catalyst 6500



The routing protocols (usually OSPF and EIGRP in a campus environment) running in global table learn routes from the routing peers and install those routes into the routing database. After the routing database has been populated, the CEF process takes the information in the database and populates the forwarding table. This table is then programmed and pushed down to the DFC (if DFC-enabled line cards are present in the system) and the PFCs on the supervisors.

In addition to this, after MPLS is enabled on the device, there is an additional control plane represented by a label distribution protocol that can be thought as a routing protocol for MPLS, because it provides neighbor devices with information about MPLS labels. The label information received from the neighbors is loaded into the label database. Once again, the CEF process running on the SP takes that information and builds a second label database. Notice that this data structure contains v4 routes, v6 routes, and MPLS forwarding entries, and those MPLS forwarding entries basically form part of it.

The commands to view the contents of these databases on the SP and DFC3s are the same as the ones used on any Cisco IOS-based distributed forwarding platform. These commands, with the relative output, are as follows:

- **show mpls forwarding-table**

```
cr20-6500-1#sh mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.100.19/32 0          Te1/1      10.122.5.30
17     Pop tag     10.122.5.10/31    0          Te1/2      10.122.5.26
       Pop tag     10.122.5.10/31    0          Te1/1      10.122.5.30
18     Pop tag     10.122.5.6/31     0          Te1/2      10.122.5.26
<SNIP>
```

- **show ip cef**

```
cr20-6500-1#sh ip cef
Prefix              Next Hop            Interface
0.0.0.0/0           10.122.5.26         TenGigabitEthernet1/2
0.0.0.0/32          receive
2.2.2.2/32          receive
10.122.5.2/31       10.122.5.26         TenGigabitEthernet1/2
                    10.122.5.30         TenGigabitEthernet1/1
<SNIP>
```

To show the platform-specific hardware databases programming, use the following commands:

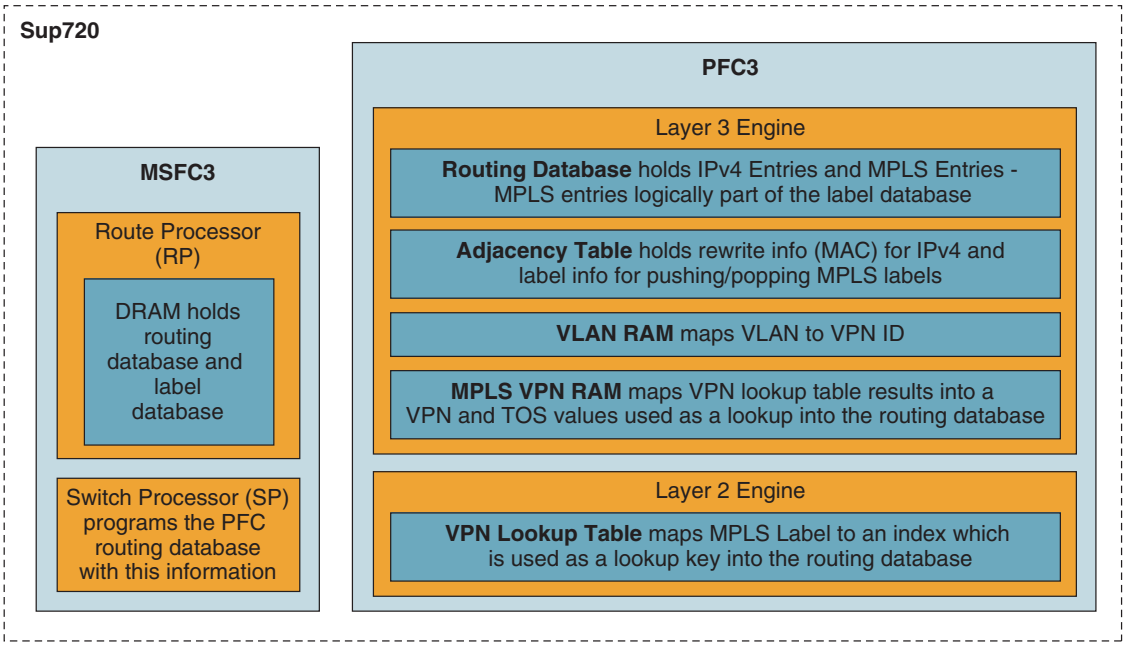
- **show mpls platform forwarding-table** (issued on PFC3 / DFC3 modules)
- **show mls cef mpls**

```
cr20-6500-1#sh mls cef mpls
Codes: + - Push label, - - Pop Label         * - Swap Label
Index  Local     Label                  Out i/f
       Label      Op
576    0   (EOS) (-)                    recirc
608    100       (-)                    Vl355         , 0009.e845.4fff
609    101       (-)                    Vl355         , 0009.e845.4fff
610    97        (-)                    Te1/3         , 0009.448e.0e00
611    98        (-)                    Vl305         , 0009.e845.4fff
<SNIP>
```

From a data plane perspective, the information in the label database is used to make that forwarding decision for outgoing MPLS packets.

Figure 45 shows the Catalyst 6500 hardware components.

Figure 45 Sup720 Architecture



There are RP and the SP processors on the MSFC3. The DRAM on the RP holds the routing and label databases. As previously discussed, the SP takes the information contained in these tables and programs the unified routing database on the PFC3. The PFC3 can be divided in two main components: Layer 3 and Layer 2 Engines. The Layer 3 Engine hosts the routing database and the adjacency table that holds rewrite information for each prefix contained in the database. Also, the Layer 3 Engine has two additional special pieces of memory, a VLAN RAM and an MPLS VPN RAM. Describing how the various label operations (PUSH, SWAP, and POP) are performed clarifies what roles each of these components need to play.

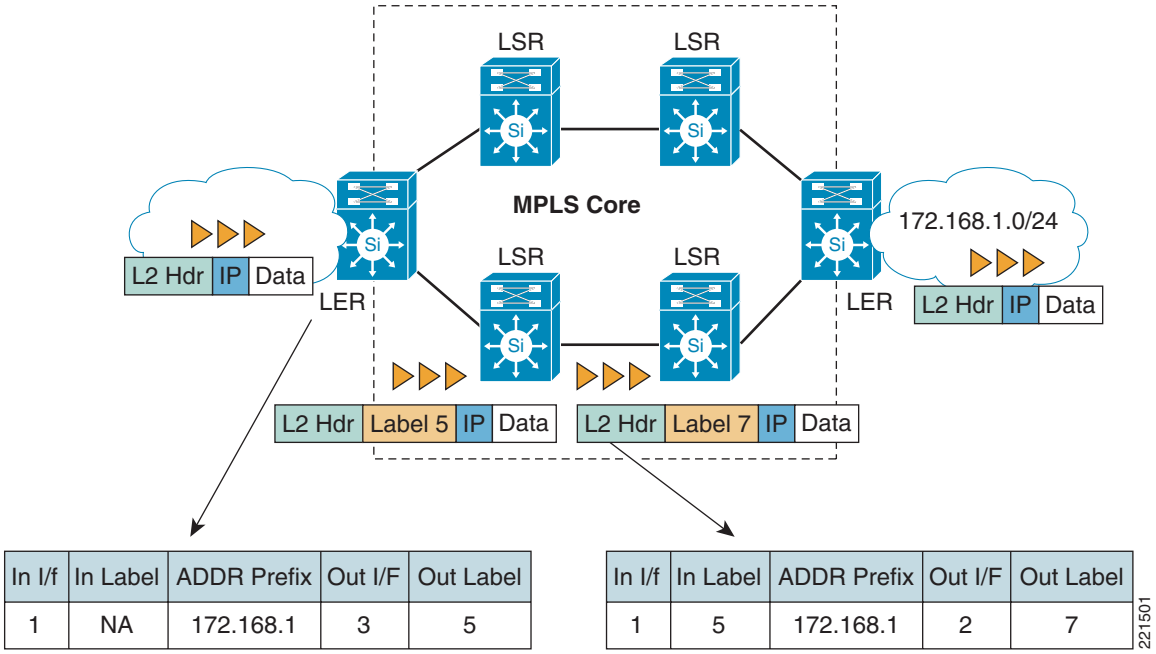
The Layer 2 Engine hosts a VPN Lookup table, which actually maps each MPLS label to an index that is used as a lookup key into the routing database. This is a key element when describing the POP operation for aggregate labels.

Note The hardware architecture described here is valid for both Sup32 and Sup720 (the PFC on the Sup720 is identical to the one on the Sup32). However, note that the MPLS functionality is supported on supervisors equipped with PFC3B and higher.

LSR and LER Defined

Depending on the specific role that the Catalyst 6500 devices play in the MPLS network, there is a distinction between a label edge router (LER) and a label switch router (LSR). (See Figure 46.)

Figure 46 LER and LSR



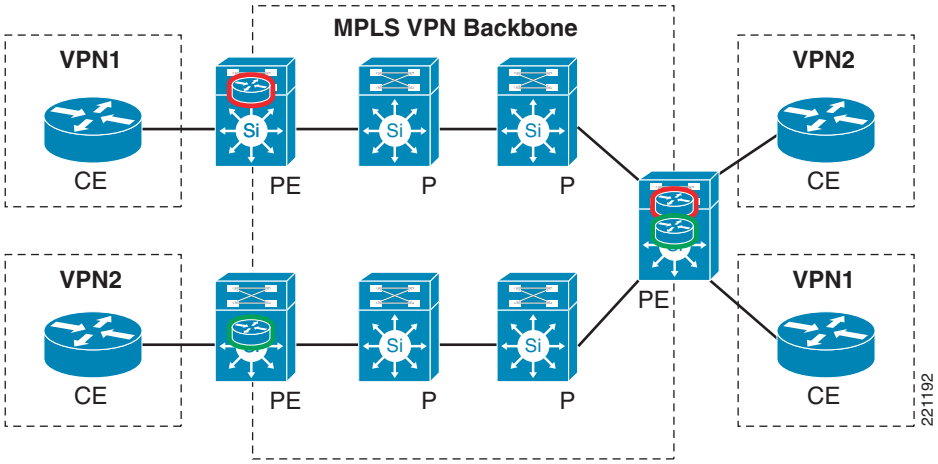
Typically, the LER sits at the edge of the MPLS cloud at the boundary between the MPLS cloud and a non-MPLS network. Its functions are to add MPLS labels to the packet as it goes into an MPLS cloud (PUSH operation), or to strip those labels off when the packet leaves the MPLS cloud and goes into the non-MPLS network (POP operation). In Figure 46, the LER receives a packet destined to the subnet 172.168.1.0/24, performs the lookup in the routing database, and pushes a specific MPLS label (label 5) to the packet before sending it toward the neighbor LSR.

The LSR is responsible for making a forwarding decision based on the outer MPLS label contained in the packets received. Referring again to Figure 46, the LSR performs a lookup in the label database and determines that a packet received on the specific interface 1 with label 5 should be switched out interface 2 with a new label 7 (SWAP operation).

Note Depending on the specific application enabled in the MPLS network (FRR, CsC, Traffic Engineering, and so on), LSR may also add labels as well, effectively creating tiers of a network hierarchy. These are usually unnecessary functions for solving the design problems in a campus MPLS VPN deployment and are not discussed further. For more information, see the following URL: http://www.cisco.com/en/US/partner/products/ps6557/products_ios_technology_home.html

Note With MPLS terminology, in addition to LER and LSR, there is often reference to three additional acronyms: P, PE, and CE (see Figure 47). They are typically used when starting to deploy VPN services over the MPLS network, and are inherited from the service provider world.

Figure 47 CE, PE, and P Devices in MPLS VPN



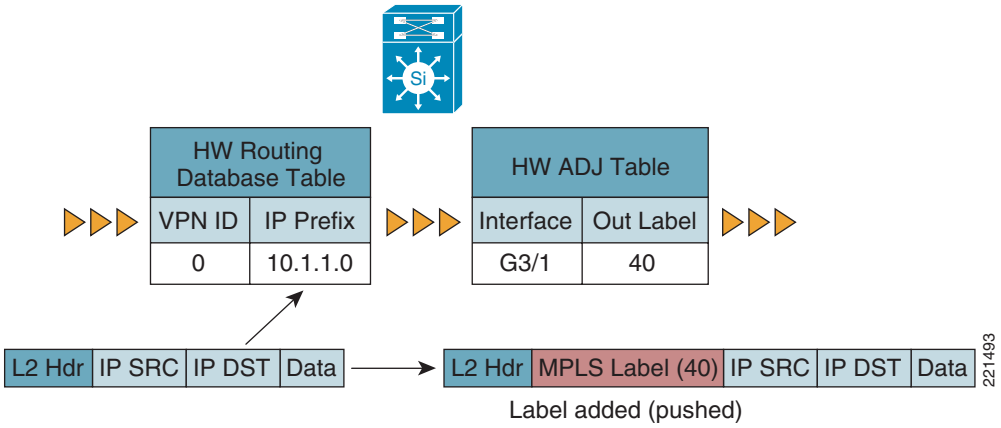
Customer edge (CE) refers to a device that sits outside of an MPLS network (traditionally at the customer site). The provider edge (PE) device is akin to the LER, whereas the provider (P) device sits inside the MPLS cloud. MPLS VPN binds the VRF-lite technology with MPLS to provide virtualization capability, using MPLS labels to make the forwarding decisions. This basically means that now LERs have to "push" two MPLS labels on each IP packet entering the MPLS cloud: one internal label (called VPN label), and one external label (called IGP label). As previously mentioned, the deployment of MPLS VPN in multilayer campus networks is characterized by the absence of CE devices, and the PEs (LERs) sitting at the distribution layer impose two MPLS labels for traffic originated from directly connected networks belonging to specific VPNs.

The following sections discuss in more detail the specific operations the Catalyst 6500 hardware needs to perform in each of the phases described above, both for simple MPLS and MPLS VPN scenarios.

LER IPv4 Routing

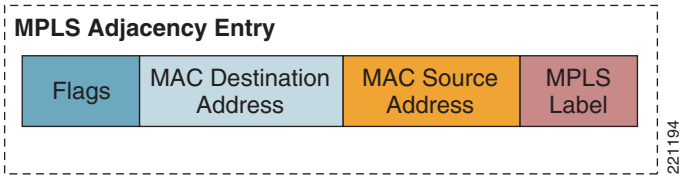
IPv4 packets are forwarded across an MPLS network by the LER that is imposing labels. After the LER imposes the label, all nodes in the MPLS network forward the packet based on the top label. The label imposed on the IPv4 packet is based on IPv4 prefix. Figure 48 illustrates an LER receiving the packet and doing a lookup in the hardware tables (routing database and adjacency), and determining that label 40 is to be used to forward the packet. The LER transmits the packet with label 40 and the relevant L2 headers for the media. The VPN ID in the CEF table is zero to indicate the global routing table.

Figure 48 LER IPv4 Routing



When acting as the ingress LER, the IPv4 packet is looked up like a regular IPv4 lookup. Because the ingress LER needs to start tagging the IP packets before sending them to the MPLS-enabled network, the adjacency entry for the IPv4 prefix needs to specify the label(s) to be imposed on the packet, as shown in Figure 49.

Figure 49 MPLS Adjacency Entry



Note Only IPv4 unicast packets have MPLS labels imposed upon them; IPv4 multicast packets are sent unlabeled.

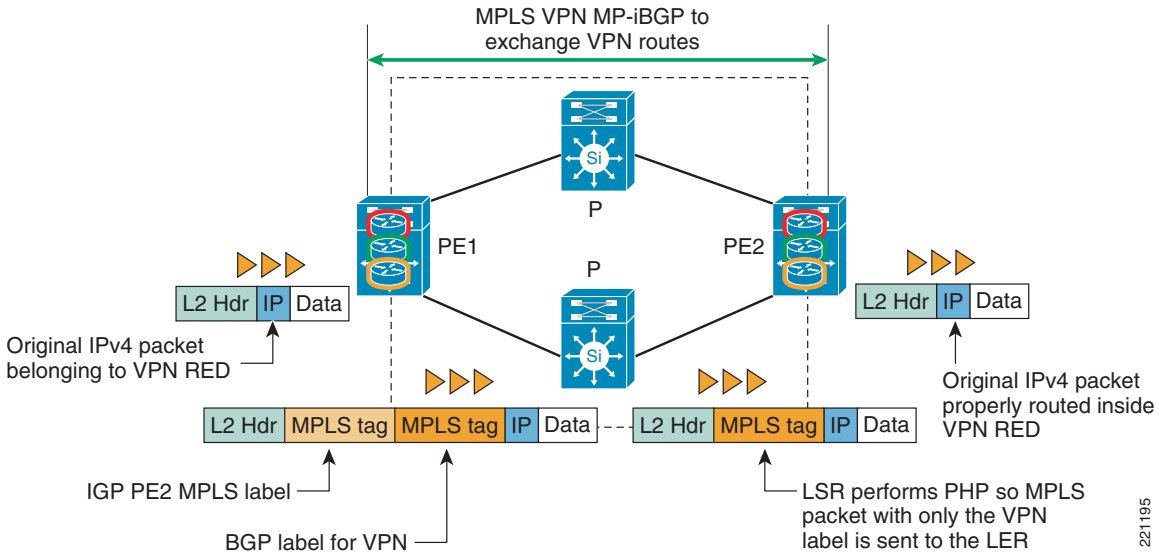
The LER device sitting at the egress edge of the MPLS cloud must remove all labels and perform an IPv4 forwarding decision on the packet (assuming it is not performing other functionalities not applicable in this design context, such as inter-AS or CsC function, in which case the behavior could involve leaving one or more labels on the packet). In most instances, the LSR device preceding the LER has popped the outermost label (PHP), and the LER receives the packet unlabeled. This is also the default behavior for Catalyst 6500 platforms, so the assumption is that the egress LER simply has to perform the forwarding decision based on the exposed IPv4 packet information.

LER IP VPN

RFC 2547 describes the implementation of L3 VPNs using BGP to distribute the VPN information between LERs (PEs). The LERs are responsible for maintaining a separate routing table for each VPN. Packets are forwarded by looking up the prefix in the VPN forwarding table, and pushing the VPN label to identify the particular VPN and the IGP label that corresponds to the BGP next hop address for the destination LER.

RFC2547 defines any-to-any connectivity model inside each defined VPN. Each VPN has a unique CEF table on a PE device; this potentially allows for VPNs to have overlapping addresses. As shown in Figure 50, PE-1 determines that the packet is destined to PE-2 by looking up the VPN table, and pushes two labels upon the packet.

Figure 50 LER and LSR Operation



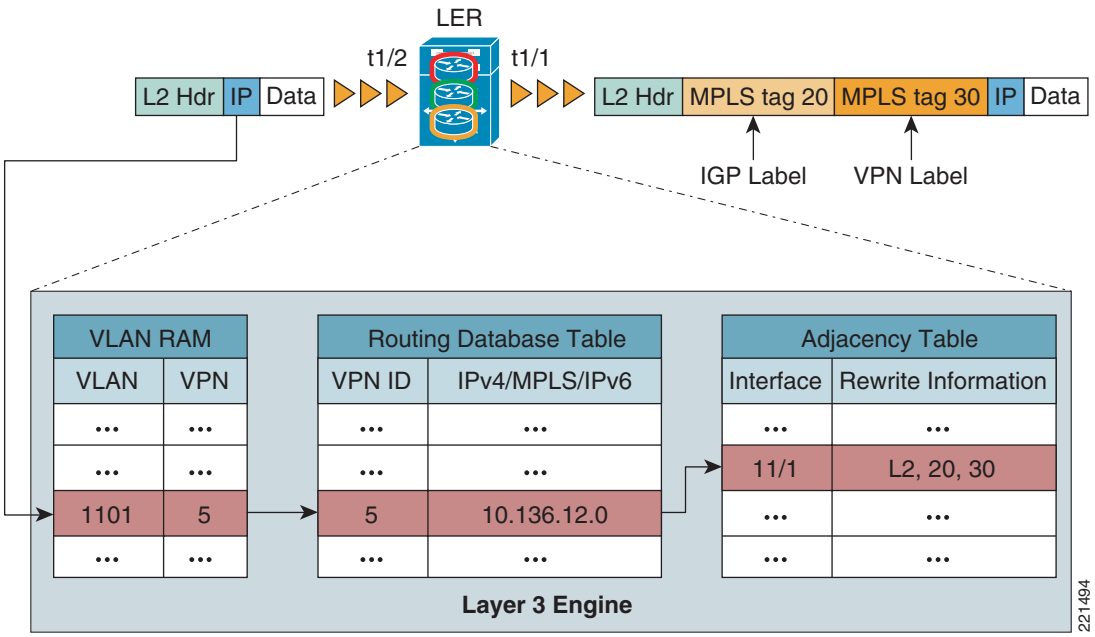
The first label pushed is a label to identify the specific VPN (VPN RED) for the PE-2. The label to be used was learned across the MP-iBGP session between PE-1 and PE-2. The second label pushed onto the packet is the IGP label to forward traffic to PE-2 along a dynamically-built LSP. By default, the last LSR connecting to PE-2 performs the PHP functionality, so PE-2 receives the packet with only the VPN label remaining. PE-2 pops the labels and performs an IP lookup on the backup to forward the packet to the destination (belonging to the proper VPN RED).

LER functionalities are performed on Catalyst 6500 platforms that are capable of hardware MPLS VPN traffic forwarding in two ways: ingress LER and egress LER.

Ingress LER

Figure 51 illustrates how the PFC3/DFC3 performs the forwarding decision for packets entering into a specific VPN. The packet is received on the interface and the headers are sent to the PFC3 to make the forwarding decision.

Figure 51 Ingress LER Operation



The following takes place:

- The Catalyst 6500 Layer 3 Engine contains a table that maps VLANs to VPNs, called VLAN RAM. The packet ingresses a specific interface (Gig 1/1 in this example) that maps to an internally allocated VLAN 1101. Every Layer 3 interface in the system has a VLAN associated with it, either by configuration ("interface VLAN"), or by internal allocation ("interface Gigabit 1/1"). Sub-interfaces also have internal VLANs allocated.

  By default, internal VLANs are assigned starting from the value 1006, as shown in the following example:

```
cr20-6500-1#sh vlan internal usage
VLAN Usage
---- --------------------
392  GigabitEth ernet2/8.392
402  GigabitEthernet2/8.402
1006 online diag vlan0
1007 online diag vlan1
1008 online diag vlan2
<SNIP>
```

  This implies that when trying to define a new L2 user VLAN, a message can be displayed to indicate that the specific VLAN is not available because it has already been internally allocated, as shown in the following example:

```
cr20-6500-1(config)#vlan 1006
cr20-6500-1(config-vlan)#name user_defined_VLAN
cr20-6500-1(config-vlan)#exit
% Failed to create VLANs 1006
VLAN(s) not available in Port Manager.
```

  To minimize this occurrence, the default behavior of the Catalyst 6500 can be changed with the command **vlan internal allocation policy descending**. This instructs the switch to allocate VLANs for internal usage starting from the highest value (4094) instead that from the lowest (1006), as in the following example:

```
cr20-6500-1#sh vlan internal usage
VLAN Usage
---- --------------------
392  GigabitEthernet2/8.392
402  GigabitEthernet2/8.402
<SNIP>
4092 online diag vlan2
4093 online diag vlan1
4094 online diag vlan0
```

Note After entering the command above, a reload of the box is required for the new VLAN allocation to become effective.

- The IP destination address is looked up in the CEF table but only against prefixes that are in the specific VPN; in the example, this is VPN number 5. The CEF table entry points to a specific set of adjacencies. One is chosen as part of the load balancing decision if multiple parallel paths exist (see Redundancy and Traffic Load Balancing, page 118 for more details on multi-path scenarios).
- The adjacency table contains the information on the L2 header the packet needs, and the specific MPLS labels to be pushed onto the frame; in the example, these are labels 20 and 30. The adjacency table can push up to three labels without the need for re-circulation (two labels are required for the MPLS VPN deployment discussed in this guide). The information to rewrite the packet is sent back to the ingress line card, where it is rewritten by the port/fabric ASICs and forwarded to the egress line interface; in this example, g1/2.

All the information shown in Figure 51 can be accessed via the CLI of the Catalyst 6500. In the following example, the packet is received on an interface mapped to VRF "v1" and is destined to a remote VPN subnet 10.136.12.0. It is possible to immediately get the information on which interface the packet will be sent out and with which labels by using the following command that accesses the content of the hardware routing table:

```
cr20-6500-1#sh mls cef vrf v1 10.136.12.0
Codes: decap - Decapsulation, + - Push Label
Index  Prefix              Adjacency
3466   10.136.12.0/24      Te1/1           313(+),57(+)
```

The output above reveals that the packet is going to be sent out interface Te1/1 with two MPLS labels: an internal VPN label (313) that is used by the receiving PE to route the traffic to the right VRF, and the external label (57) that is used to label switch the traffic along the LSP connecting the ingress LER to the egress LER (this is also shown in Figure 51).

Note The symbol "+" associated to the MPLS tag in the output above indicates that these labels are going to be pushed to the packet.

You can retrieve detailed hardware information for the same VPN destination prefix by using the following command:

```
cr20-6500-1#sh mls cef vrf v1 10.136.12.0 detail
Codes: M - mask entry, V - value entry, A - adjacency index, P - priority bit
       D - full don't switch, m - load balancing modnumber, B - BGP Bucket sel
       V0 - Vlan 0,C0 - don't comp bit 0,V1 - Vlan 1,C1 - don't comp bit 1
       RVTEN - RPF Vlan table enable, RVTSEL - RPF Vlan table select
Format: IPV4_DA - (8 | xtag vpn pi cr recirc tos prefix)
Format: IPV4_SA - (9 | xtag vpn pi cr recirc prefix)
M(3466   ): E | 1 FFF  0 0 0 0   255.255.255.0
V(3466   ): 8 | 1 256  0 0 0 0   10.136.12.0        (A:278534 ,P:1,D:0,m:0 ,B:0)
```

Two important pieces of information can be retrieved from the output above:

- The pointer to the adjacency table containing the rewriting information (A:278534)
- The number of equal cost paths available to reach the destination prefix (P:1, which means there is only one path in this example)

Using the information above, you can then access the corresponding entry in the adjacency table, as follows:

```
cr20-6500-1#sh mls cef adjacency entry 278534 detail
Index: 278534  smac: 0012.da7c.c680, dmac: 0004.de1f.b000
               mtu: 1526, vlan: 1035, dindex: 0x0, l3rw_vld: 1
               format: MPLS, flags: 0x8418
               label0: 0, exp: 0, ovr: 0
               label1: 313, exp: 0, ovr: 0
               label2: 57, exp: 0, ovr: 0
               op: PUSH_LABEL2_LABEL1
               packets: 0, bytes: 0
```

The output shows the rewrite information for the packet: source MAC, destination MAC, and the MPLS labels that are pushed to the packet (57 and 313). Also, the internal VLAN is reported (VLAN 1035), which maps directly to the interface that is used to forward the packet. It is already known that the interface used is Te1/1, and this is confirmed by displaying the mapping between internal VLANs and interfaces:

```
cr20-6500-1#sh vlan internal usage
VLAN Usage
---- --------------------
1006 online diag vlan0
1007 online diag vlan1
......................
1035 TenGigabitEthernet1/1
1036 TenGigabitEthernet1/3
.........................
```

Egress LER

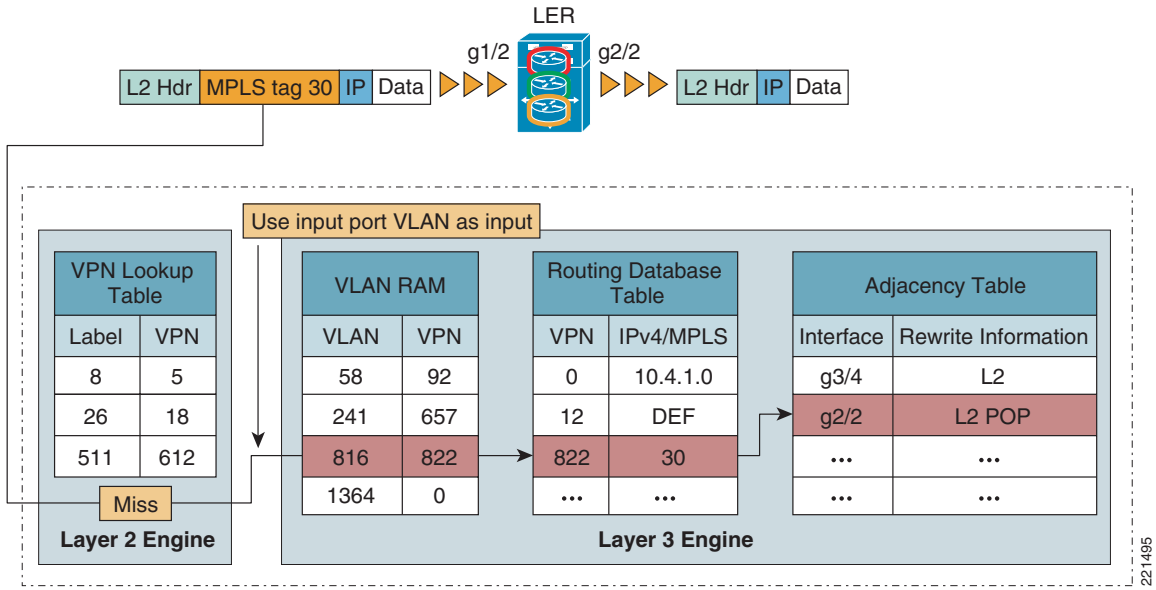
The way the PFC3/DFC3 handles VPN traffic on egress from the PE varies depending on whether per-prefix labels or aggregate labels are used. When per-prefix labels are used, each VPN prefix has a unique label association, which allows the PE to forward the packet to the final destination based on a label lookup in the routing database. If aggregate labels are used, the PFC3/DFC3 must perform an IP lookup to determine the final destination because many prefixes that can be on multiple interfaces are associated with the same label. Note that aggregate labels are assigned to each directly connected subnet, or every time a device performs route summarization.

It is important to note that when deploying MPLS VPN in a multilayer campus environment, positioning the PE at the distribution layer implies that all the VPN subnets result directly connected to the PE device. The PE then assigns a unique aggregate label to each defined VRF; this is to allow it to properly perform the lookup in the right routing table for all the VPN traffic received from the core of the network. In the following example, there is a specific PE assigning an unique aggregate label to each locally defined VRF (there are 25 VRFs in this case).

The implication of using aggregate labels is subsequently discussed in more detail.

Figure 52 illustrates the egress processing by PFC3/DFC3 when per-prefix labels are used.

Figure 52 Egress LER Operation with Per-Prefix Labels

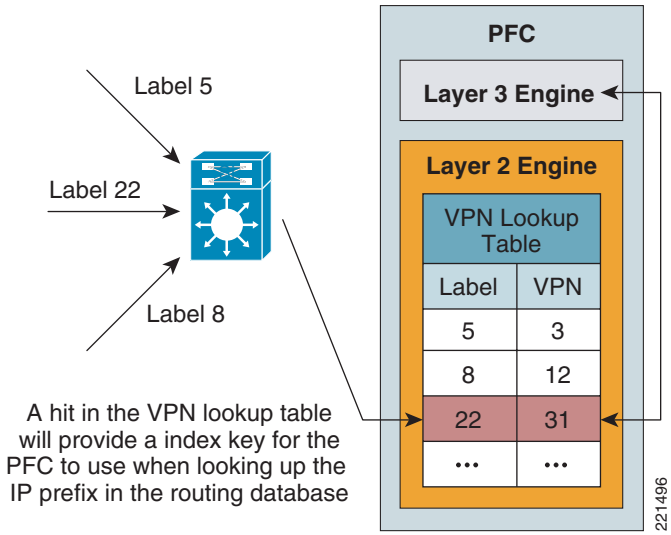


The sequence of events that happen for performing the popping of a per-prefix label is the following:

1. The packet enters the switch on a given interface (for which the switch assigns an internal VLAN number, 816 in this example). The MPLS label 30 present on the packet represents the VPN label, because by default the previous node in the network has performed PHP to remove the external IGP label.
2. The packet headers are sent from the line card to the PFC3/DFC3 complex to perform the forwarding decision. The VPN label (30) does not match an entry in the VPN lookup table hosted in the Layer 2 Engine ("MISS" event). This is because, as discussed further below, the VPN lookup table is used only to store aggregate labels.
3. As a consequence, the packet headers are sent to the Layer 3 Engine and a lookup is performed in the VLAN RAM table using the internal VLAN index associated to the port of the switch that received the packet (816 in this example). The lookup in the VLAN RAM determines that the packet belongs to the VRF identified by the VPN ID 822.
4. This information is used to look up the MPLS label in the routing table (associated to the specific VPN ID). The appropriate adjacency is then chosen after performing the load balancing hash if multiple parallel paths exist. The adjacency contains the outbound interface (Gig 2/2) and L2 headers and tells the system to POP the last label and to forward the packet to the next hop/ destination as an IP packet.

Figure 53 shows how the Catalyst 6500 performs the pop operation when the packet contains an aggregate MPLS label. As mentioned before, unique aggregate labels are assigned to each VRF defined on the PE device; aggregate labels are stored in the VPN lookup table, which is a table hosted on the Layer 2 Engine of the PFC3.

Figure 53 POP Operation with Aggregate Label



In Figure 53, Label 5, Label 8, and Label 22 are aggregate labels and are stored in the VPN lookup table table. The other information in the table associated to each aggregate label is the VPN ID that is used as part of the lookup key into the routing database. The important thing to consider here is that the VPN lookup table can host at most 512 entries. Allocating more than 512 aggregate labels on the PE device results in recirculation, thus reducing switching performance. Because a unique aggregate VPN label is associated to each VRF defined on the egress PE device, the number 512 represents the maximum number of VRFs that should be defined on a given PE to achieve optimal performance. This is rarely an issue in campus MPLS VPN deployments.

Note One entry in the VPN lookup table is always reserved for the Explicit NULL label; therefore, the optimal performance is actually achieved with a maximum of 511 aggregate labels.

Information about the current usage of the VPN lookup table can be retrieved with the following CLI command:

```
cr20-6500-1#sh platform hardware capacity pfc
L2 Forwarding Resources
           MAC Table usage:   Module  Collisions  Total       Used       %Used
                              1                0  65536         94         1%
                              2                0  65536        105         1%
                              5                0  65536         94         1%
             VPN CAM usage:                       Total       Used       %Used
                                                    512         25         5%
```

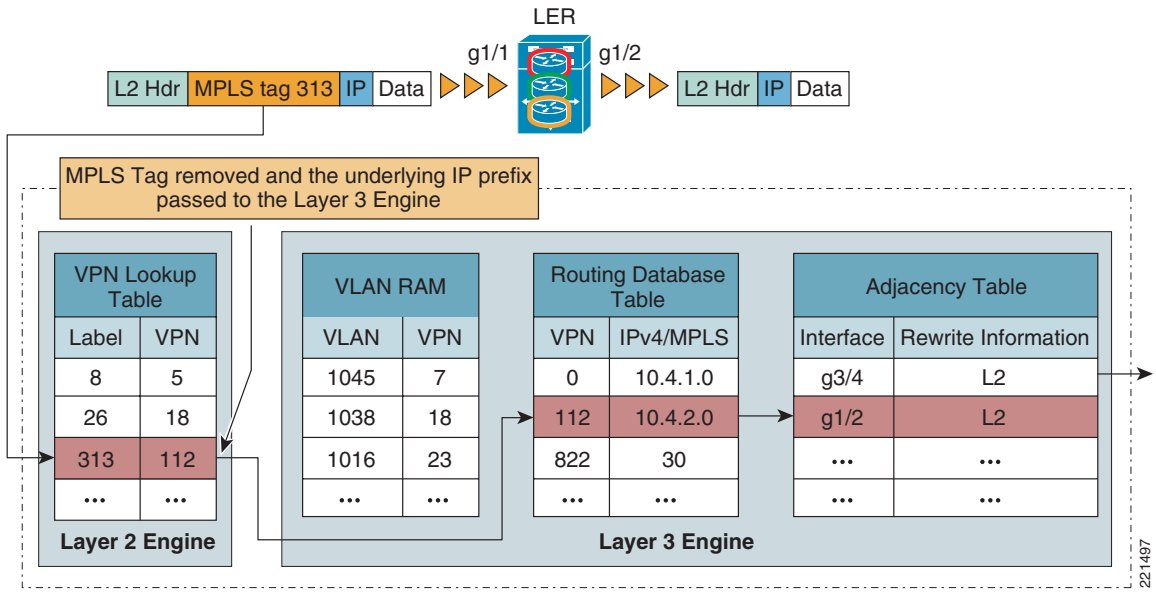
The example above refers to a PE that has allocated 25 aggregate labels for each distinct locally defined VRF, as follows:

```
cr20-6500-1#sh mpls forwarding-table | i Aggregate
44     Aggregate   vrf:v0160
65     Aggregate   vrf:v0260
66     Aggregate   vrf:v030
67     Aggregate   vrf:v040
68     Aggregate   vrf:v0560
69     Aggregate   vrf:v060
70     Aggregate   vrf:v070
71     Aggregate   vrf:v080
72     Aggregate   vrf:v090
```

```
73      Aggregate   vrf:v100
74      Aggregate   vrf:v110
75      Aggregate   vrf:v120
76      Aggregate   vrf:v130
77      Aggregate   vrf:v140
78      Aggregate   vrf:v150
79      Aggregate   vrf:v160
80      Aggregate   vrf:v170
81      Aggregate   vrf:v180
82      Aggregate   vrf:v190
83      Aggregate   vrf:v200
84      Aggregate   vrf:v210
85      Aggregate   vrf:v220
86      Aggregate   vrf:v2360
87      Aggregate   vrf:v2460
88      Aggregate   vrf:v250
```

Depending on whether the number of aggregate labels is more or less than 512, the pop operation would happen in a different way. Figure 54 shows the scenario where the number of aggregate labels is less than 512.

Figure 54 POP Operation with Less than 512 Aggregate Labels



In Figure 54, the following sequence takes place:

1. The packet is received on the egress LER with only the VPN label (the previous node in the network performed PHP to remove the IGP label).
2. The packet headers are sent from the line card to the PFC3/DFC3 complex to perform the forwarding decision. The VPN label (313) matches an entry in the VPN lookup table and this allows for the Layer 2 Engine to determine the VPN ID (112) for the specific packet and to pop the VPN label. This allows the Layer 2 Engine to process the packet as an IP packet in a single pass without having to first pop the MPLS label and then re-circulate the packet to process it in the second pass as an IP packet.
3. The result from the VPN lookup table is sent with the packet IP headers to the Layer 3 Engine. Note that the VLAN RAM table is not used to determine the VPN ID when a hit occurs in the VPN lookup table.

4. The IP destination address (10.4.2.0) is looked up in the routing database against the routes for VPN 112. The appropriate entry in the adjacency table is then chosen after performing the load balancing hash if multiple parallel paths exist. The adjacency contains the outbound interface (Gig 1/2) and L2 headers to forward the packet to the next hop/destination.

Note In the procedure described above, the processing of the packet happens in a single pass without the need for any hardware recirculation. This explains why optimal system performances are achieved in this case.

Figure 55 shows a different scenario where the VPN lookup table is full because more than 512 aggregate labels were allocated on this given PE.

Figure 55 POP Operation with More Than 512 Aggregate Labels

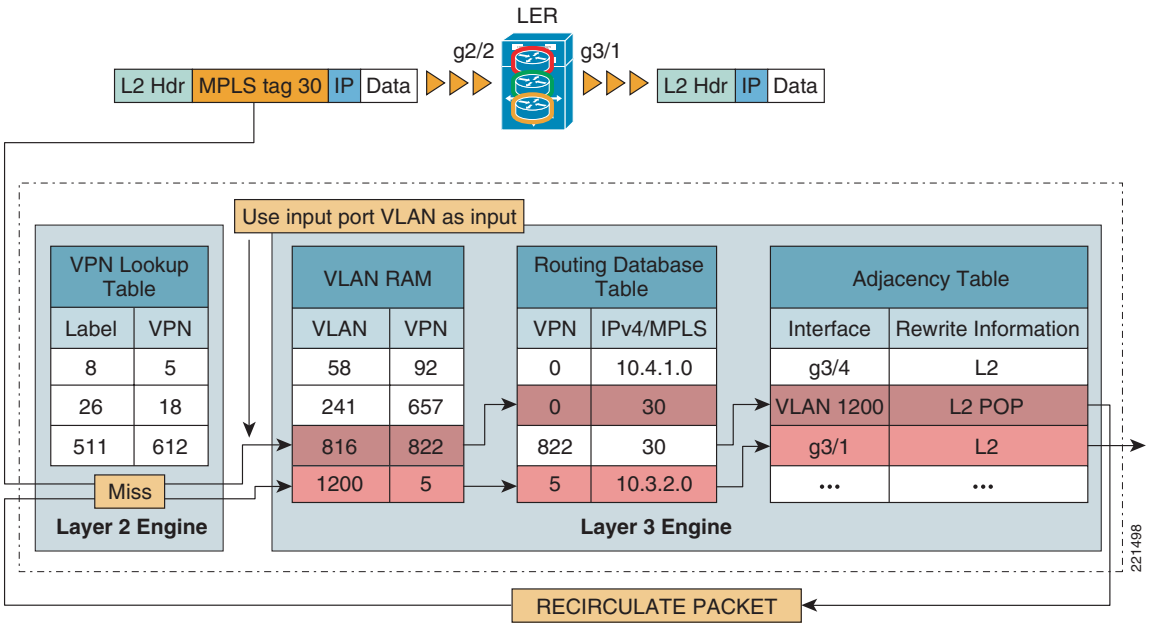


Figure 55 illustrates the egress processing by PFC3/DFC3 when the VPN number is greater than 512 and an aggregate label is being used. The following sequence takes place:

1. The packet enters the switch with the VPN label.
2. The packet headers are sent from the line card to the PFC3/DFC3 to perform the forwarding decision. The VPN label (30) does not match an entry in the VPN lookup table, because the table is full and in this example, label 30 is not part of it. This causes the Layer 2 Engine to send the packet to the Layer 3 Engine as an MPLS packet; this is because the MPLS label information is required to perform the routing database lookup at the following step.
3. The Layer 3 Engine receives the packet and performs the VLAN to VPN mapping that result in VPN 0 being selected. The label (30 in this example) is then looked up in the CEF table and the correct adjacency selected. The adjacency indicates that the MPLS label is to be popped and then the packet re-circulated on internal VLAN 1200.
4. The packet is sent back to the rewrite engine associated with the particular port and rewritten. The packet then arrives in the Layer 2 Engine the second time and hits a “MISS” in the VPN lookup table (this time because it is an IP packet with no MPLS label information).

5. The IP packet is passed to the Layer 3 Engine and the VLAN RAM table determines that the packet belongs to VPN 5 (using the internal VLAN 1200 information applied to the packet before recirculation).
6. The destination address is then looked up in the CEF table against the routes for VPN 5. The appropriate adjacency is then chosen after performing the load balancing hash if multiple parallel paths exist. The adjacency contains the outbound interface (Gig 2/2) and L2 headers to forward the packet to the next hop/destination.

Therefore, for those situations where there are more than 512 VPNs, packet recirculation is required, which means two passes through the PFC, and the entire performance of that particular packet as part of that MPLS VPN drops.

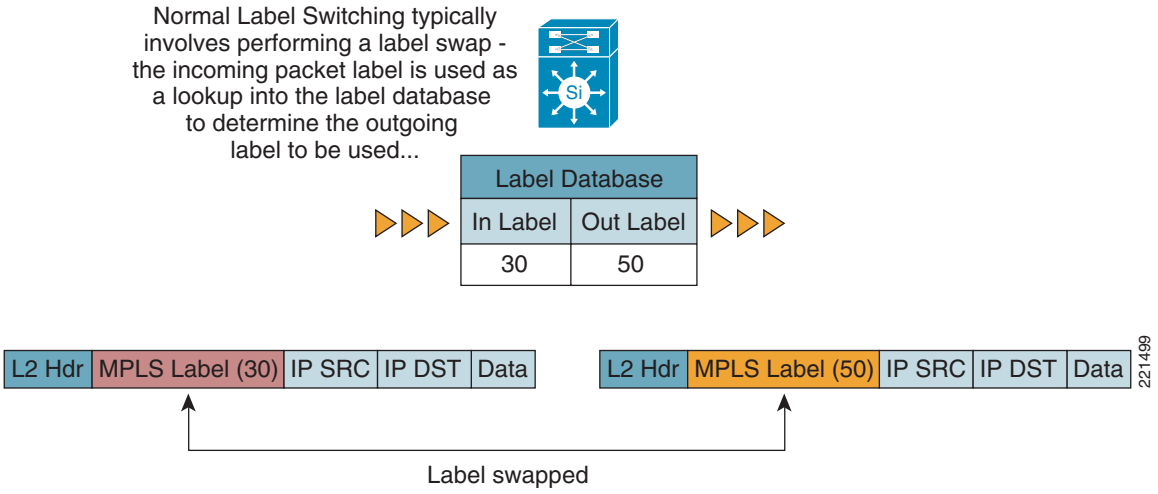
In summary, when performing egress PE functionalities on a Catalyst 6500, optimal performances are achieved only when the number of VRFs defined on the specific PE devices is less than 512; this is not a big issue for campus deployment, where rarely the number of required VPNs is higher that 50. In addition, even when deploying more than 512 VRFs, the performances are reduced only for traffic belonging to the VRFs defined from 513 and above.

LSR Functionality

LSRs receive labeled packets and, depending on their position in the MPLS network, can perform a swap or pop operation. A swap operation is required when the packet comes in with a label and needs to be forwarded to another LSR; in this case, the original label is exchanged with a new label that represents the label this node uses to reach the ultimate destination.

As shown in Figure 56, to perform label swapping, the LSR uses the incoming packet label to execute the lookup into the hardware label database and to determine the new label that should be pushed to the packet before sending it to the neighbor LSR.

Figure 56 LSR Functionality

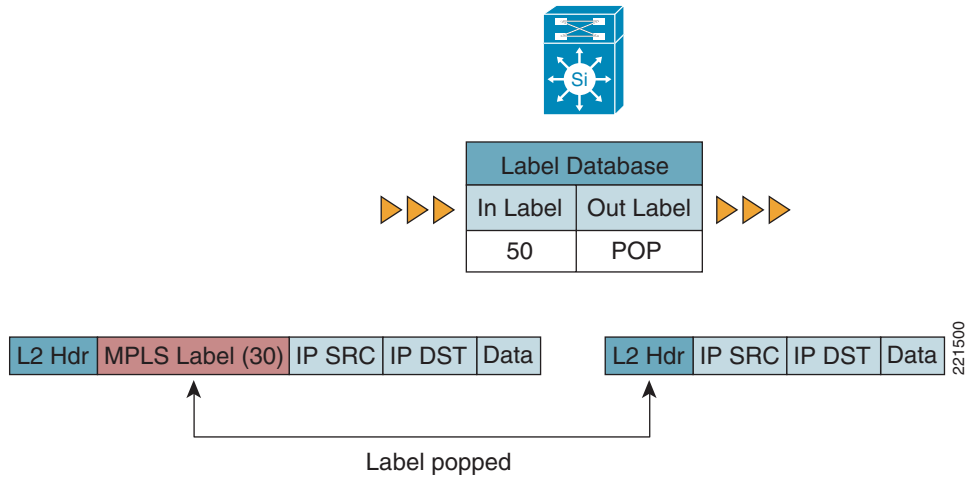


Note VPN traffic is characterized by having two MPLS labels added to the packet. However, the label switching is performed by the LSR, always based on the outer label.

The pop operation occurs if this node is performing PHP. If the LSR is adjacent to LER, it is standard behavior to remove the outermost label before forwarding the packet to the LER. This makes the forwarding decision on the LER simpler. For example, in the case of IPv4 unicast, the LER has to perform only an IP forwarding decision instead of a label and IP lookup.

As shown in Figure 57, the information to perform the POP operation is again contained in the hardware label database.

Figure 57 POP Operation

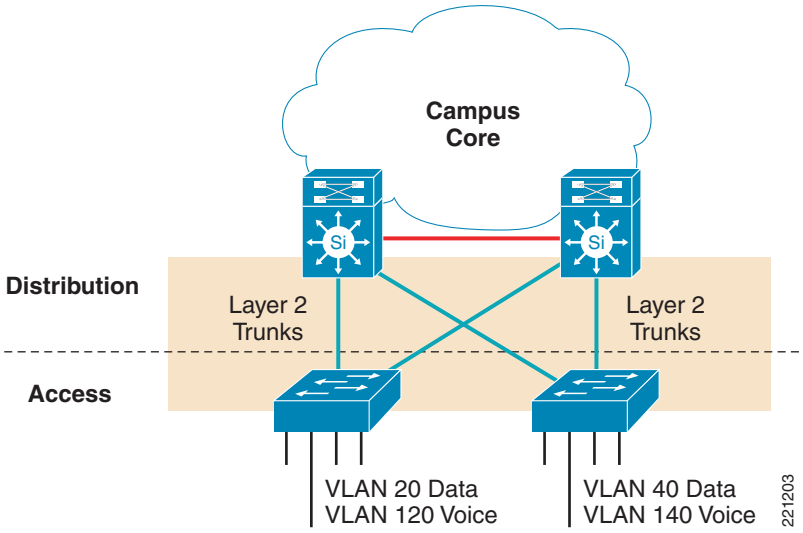


Note The example in Figure 57 refers to normal MPLS traffic. As discussed in the previous sections, in case of VPN traffic, the packet sent from the penultimate hop device toward the egress LER also contains the VPN label.

Virtualizing the Campus Distribution Block

In the traditional multilayer campus design, the access layer is deployed with L2 capabilities only, and the distribution layer devices represent the boundary between L2 and L3 domains in the network. The generic campus distribution block is shown in Figure 58.

Figure 58 Campus Distribution Block

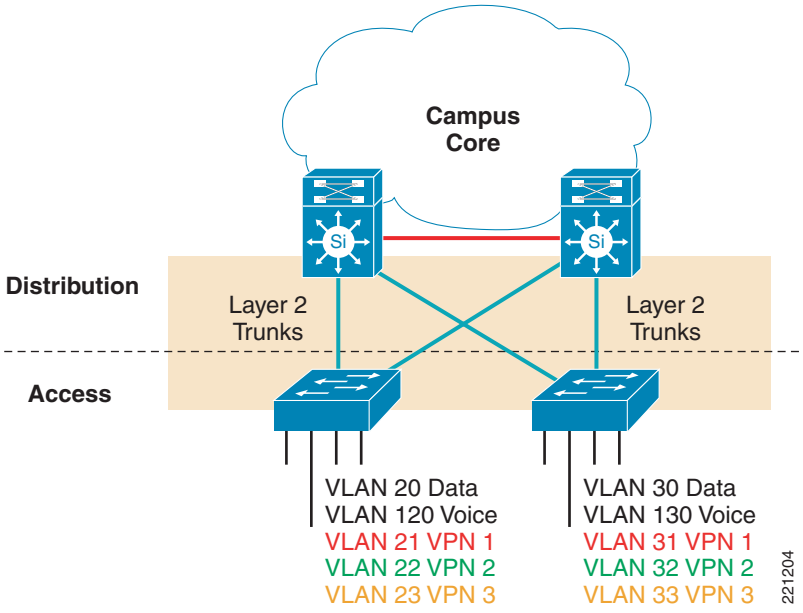


More details on the recommended configuration and deployment guidelines for the traditional distribution block design can be found in the campus design guides previously referenced in this guide. This section discusses design choices and the impact of deploying network virtualization (specifically referring to MPLS VPN) on the design of each single campus distribution block.

VLANs and VRFs Definition

The first thing to consider is that virtualization at L2 is nothing new and is still achieved by using VLANs. As a consequence, the network virtualization requirement of supporting different logical groups in the same campus network drives the definition of an additional number of VLANs in each access layer device; these VLANs are then carried upstream toward the distribution layer via L2 trunk connections. (See Figure 59.)

Figure 59 VLAN Definition



In addition to the previous existing data and voice VLANs, new VLANs (at least one per each new VPN) are now required to provide logically isolated access to separate groups. The following additional considerations are required:

- The deployment of various users into their corresponding segments (VLANs) can be achieved through static configuration (each edge port is manually assigned to a specific VLAN), or via dynamic mechanism such as 802.1x. This is discussed more extensively in the *Network Virtualization—Access Control 2.0 Design Guide* (OL-13634-01).
- Following the recommended design to keep VLAN numbers unique per access layer switch (as shown in Figure 59) may require the creation of a high number of VLANs (and corresponding SVIs) on the distribution layer devices. This needs to be taken into consideration especially for very large distribution layer blocks (high number of access layer switches connecting to the same distribution layer pair), because it creates the following two main issues:
  - Need for planning for new VLANs and corresponding IP subnet allocation

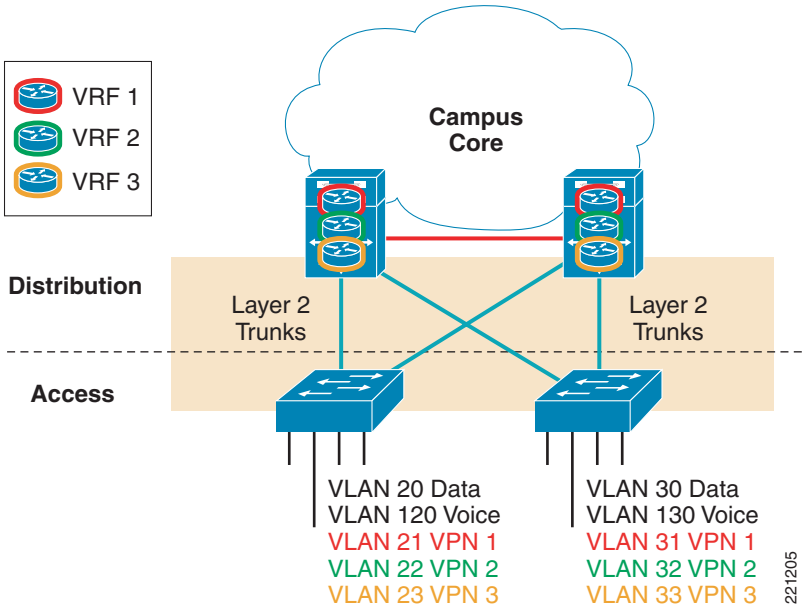
    In very large deployments, it may be required to extend the range of VLANs that can be defined on a Catalyst 6500 platform. By default, the upper limit is 1001, but it can be extended to 4094 by using the following command:

    ```
    cr20-6500-1(config)#spanning-tree extend system-id
    ```

  - Increase of the control plane load for protocols such as Spanning Tree, HSRP, and so on

The logical isolation provided by VLANs ceases to exist at the boundary between L2 and L3 domains (the distribution layer devices); it is thus required to define VRFs on these devices and map each VLAN to its own dedicated VRF instance, as shown in Figure 60.

Figure 60 VRF Definition



The idea here is to maintain the originally defined data and voice VLANs mapped to the global table (the default VRF); the reasoning behind this choice is thoroughly discussed in Path Isolation Initial Design Considerations, page 12. The other VLANs introduced to support various user groups can instead be mapped to their own VRF: this is because the virtualized network consists of the combination of the L2 VLAN and the L3 VRF, so a mapping between these components is required to achieve logical isolation end-to-end across the network. It is worth noting that, independently from the number of VLANs defined

in the campus distribution block, the number of VRFs is directly dictated by the number of logical groups that need to be supported. For example, all the VLANs defined on each access layer device for "VPN 1" users are mapped to the same "VRF 1" defined at the distribution layer.

Following is a configuration sample that demonstrates how to define VRFs and map VLANs to them on a generic distribution layer switch:

```
ip vrf VRF_1
 rd 64001:1
 route-target export 64000:1
 route-target import 64000:1
!
vlan 21
 name VPN_1_access_switch_1
!
interface Vlan21
 description VPN 1 on Access Switch 1
 ip vrf forwarding VRF_1
 ip address 10.137.21.1 255.255.255.0
```

Note The meaning of "rd" and "route-target" is clarified below when discussing the MP-BGP deployment.

After VLANs have been mapped to the corresponding VRFs, there is the need to connect together the VRFs defined in different campus distribution blocks. As previously discussed, MPLS VPN represents a dynamic solution to provide any-to-any connectivity between VRFs belonging to the same VPN. Before entering into the details of how to configure the campus Distribution block to support MPLS VPN, some considerations are provided about the virtualization of network services that are usually required in the distribution layer.

Virtualization of Network Services at the Distribution Layer

In the multilayer campus design, the distribution layer represents the network boundary between the L2 and the L3 domains. As such, there are several network services that are specifically recommended to be enabled on these network devices. Some of the services are directly related to L3 functionalities, some to L2 functionalities.

A typical example of L2 functionality is the Spanning Tree Protocol; it has already been pointed out how enabling network virtualization may result in the growth of VLANs defined in the distribution block devices. This may impact the spanning tree design, because for example there would be more instances of the protocol running (usually one per each VLAN). However, there is no requirement for adding any functionality to spanning tree, because it still works at L2 the same way it has always done.

A different case is when analyzing the L3 functionalities enabled in the distribution layer. Defining VRFs in fact allows you to virtualize the network device at L3, but this implies that all the L3 network services needs to be somehow virtualized as well (or made VRF-aware). Therefore, it is important to highlight what functionalities are available today on Catalyst 6500 platforms, pointing out also the new ones that may become available in future IOS releases of code.

Note The following list is not exhaustive, but highlights only the specific services that are discussed in the campus design guides previously referenced in this guide. Be sure to verify with the release note the VRF support for additional features that may be required in specific design cases.

First Hop Redundancy Protocol

Because the distribution layer devices represent the first L3 hop in the network, they function as the default gateway for all the clients deployed in the IP subnets belonging to the specific distribution block. Traditionally, a First Hop Redundancy Protocol (FHRP) is deployed to allow the distribution layer pair of devices to function as a single virtual device from the default gateway functionality point of view. Three protocols can usually be implemented for this:

- Hot Standby Routing Protocol (HSRP)
- Gateway Load Balancing Protocol (GLBP)
- Virtual Router Redundancy Protocol (VRRP)

The first two protocols were developed by Cisco, whereas VRRP is the IETF standard based of HSRP (RFC 3768). From a VRF awareness point of view, up to IOS release 12.2(18)SXF6, only HSRP is supported on L3 interfaces that are mapped to a specific VRF. FHRP protocols perform their functionality adding Address Resolution Protocol (ARP) entries and IP hash table entries (aliases); this by default is done using the default routing table instance. However, because a different routing table instance is used when VRF forwarding is configured on an interface, ARP and Internet Control Message Protocol (ICMP) echo requests for the HSRP virtual IP address fail, unless the protocol is made VRF-aware, thus capable of using the information in the VRF-specific routing table.

This is the case with HSRP, as shown in the following configuration samples:

- Distribution switch 1 (HSRP Active)

```
interface Vlan12
 description Users in VPN v1
 ip vrf forwarding v1
 ip address 10.137.12.3 255.255.255.0
 standby 1 ip 10.137.12.1
 standby 1 timers msec 250 msec 750
 standby 1 priority 105
 standby 1 preempt delay minimum 180
 standby 1 authentication ese
```

- Distribution switch 2 (HSRP Standby)

```
interface Vlan12
description Users in VPN v1
ip vrf forwarding v1
ip address 10.137.12.2 255.255.255.0
standby 1 ip 10.137.12.1
standby 1 timers msec 250 msec 750
standby 1 authentication ese
```

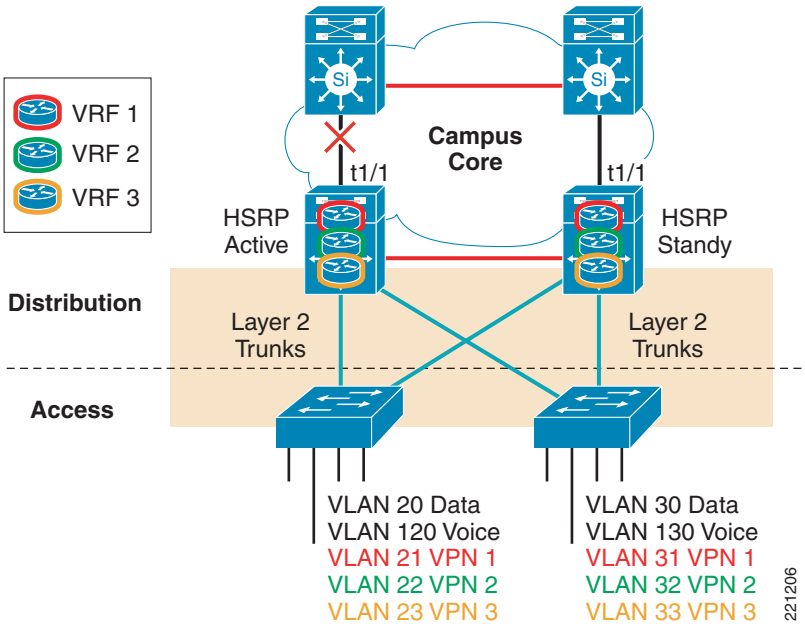
As noticed above, the configuration is essentially identical to the traditional one required on L3 interfaces belonging to global table (the default VRF). However, the VRF awareness capability allows for example to have two separate L3 VLAN interfaces with overlapping IP addresses and mapped to different VRFs (for example v1 and v2). Without VRF awareness, HSRP would get confused, whereas the capability allows the protocol to maintain a separate state for the two set of interfaces, as follows:

```
cr20-6500-1#sh standby vlan 2
Vlan2 - Group 1
  Local state is Active, priority 105, may preempt
  Preemption delayed for at least 180 secs
  Hellotime 250 msec, holdtime 750 msec
  Next hello sent in 0.033
  Virtual IP address is 10.137.12.1 configured
  Active router is local
  Standby router is 10.137.12.2 expires in 0.510
  Virtual mac address is 0000.0c07.ac01
  Authentication text "ese"
```

```
  2 state changes, last state change 00:02:37
  IP redundancy name is "hsrp-Vl2-1" (default)
cr20-6500-1#sh standby vlan 12
Vlan12 - Group 1
  Local state is Active, priority 105, may preempt
  Preemption delayed for at least 180 secs
  Hellotime 250 msec, holdtime 750 msec
  Next hello sent in 0.218
  Virtual IP address is 10.137.12.1 configured
  Active router is local
  Standby router is 10.137.12.2 expires in 0.530
  Virtual mac address is 0000.0c07.ac01
  Authentication text "ese"
  11 state changes, last state change 2d00h
  IP redundancy name is "hsrp-Vl12-1" (default)
```

One additional consideration is required for HSRP tracking; deploying HSRP tracking is usually not required or recommended in a fully redundant campus topology. However, there are some designs where it is deployed, specifically when the distribution block is not connected to the core in a fully meshed fashion, as shown in Figure 61.

Figure 61 Deploy HSRP Tracking



In this case, usually the HSRP tracking is configured so that if the interface connecting to the core fails (Ten1/1), the HSRP standby becomes active, avoiding the use of the transit link between the distribution peers for all the upstream traffic.

- Distribution switch 1 (HSRP Active)

```
interface Vlan12
 description Users in VPN v1
 ip vrf forwarding v1
 ip address 10.137.12.3 255.255.255.0
 standby 1 ip 10.137.12.1
 standby 1 timers msec 250 msec 750
 standby 1 priority 105
 standby 1 preempt delay minimum 180
 standby 1 authentication ese
```

```
standby 1 track TenGigabitEthernet1/1
```

- Distribution switch 2 (HSRP Standby)

```
interface Vlan12
description Users in VPN v1
ip vrf forwarding v1
ip address 10.137.12.2 255.255.255.0
standby 1 ip 10.137.12.1
standby 1 timers msec 250 msec 750
standby 1 authentication ese
standby 1 preempt
```

The physical interface connecting to the core (Ten1/1 in our example) usually belongs to global table; however, configuring HSRP tracking for that specific interface also for a SVIs mapped to a VRF (as shown above), allows triggering the failover also for that specific VPN subnet. The recommendation is thus to use tracking on all the SVIs defined in the distribution block (belonging to global table and to each defined VRF).

DHCP Relay

Distribution layer devices provide DHCP relay support for the endpoints connected to the access switches. Because the DHCP infrastructure is usually deployed in a centralized location in the network (for example, in a data center), this means that the first L3 hop devices need to be able to relay the initial broadcast DHCP request received from the client to the remotely located DHCP server. This is supported via the **ip helper-address** command, as shown in the following configuration sample:

```
interface Vlan12
 description Users in VPN v1
 ip vrf forwarding v1
 ip address 10.137.12.3 255.255.255.0
 ip helper-address 10.136.2.8
```

As noticed above, the **ip helper-address** command is supported also on SVIs belonging to a specific VRF. This means that the broadcast message is properly relayed in the VPN to where the interface belongs. Note that this does not actually mean that the DHCP relay functionality on Catalyst 6500 platforms is VRF-aware; to achieve VRF-awareness, the switch should be able to include VPN-specific information in the message sent to the centralized DHCP server. This would for example allow the centralized DHCP server (assuming the server is VRF-aware as well) to provide IP addresses from overlapping IP pools belonging to separate VRFs. VRF-awareness for DHCP-relay functionality is planned in an upcoming IOS release for Catalyst 6500, but assuming there is no need for overlapping IP addresses, the **ip helper-address** command currently allows to provide IP addresses to clients belonging in different VPNs.

Multicast

Multicast documentation will be available in a future design guide chapter.

QoS

QoS and network virtualization are currently orthogonal problems. Enabling VRF capabilities allows the creation of a separate control and data plane for the switch. However, there are not virtualization capabilities from a queuing perspective. This means that, for example, if traffic is classified at the edge and marked as EF, it makes use of the priority queue (if defined) independently of the origination of the VPN. Usually the distribution layer switches require the following QoS policies:

- DSCP trust policies—These are usually enabled on all the interfaces of the distribution layer device. Adding virtualization to the design does not change this requirement.

- Queuing policies—As already mentioned, queuing of the traffic is based on how the packets are classified and marked at the edge of the network. This is independent from the fact that these packets belong to a VPN or to global table (no VRF awareness is supported today for the queuing mechanism).
- Optional per-user microflow policing policies—Catalyst 6500s with PFC3 support user-based rate limiting (UBRL). UBRL is a form of Microflow policing allowing the administrator to rate limit traffic flows, but unlike a normal Microflow Policer, it allows a policer to be applied to all traffic to or from a specific user. This is independent from the VPN to which the user belongs, because the policing is usually applied on the trunk interface connecting the distribution block to the access layer device. UBRL functionality is not currently VRF-aware (that is, it is not possible to differentiate traffic from users having the same IP address but belonging to different VPNs).

Routed ACLs

Standard and extended ACLs are usually applied as routed ACLs (RACLs) at the first L3 hop of the network and have been made VRF-aware since release 12.2(18)SXD. This means they can be successfully applied to L3 interfaces (usually SVIs) that are part of a specific VRF.

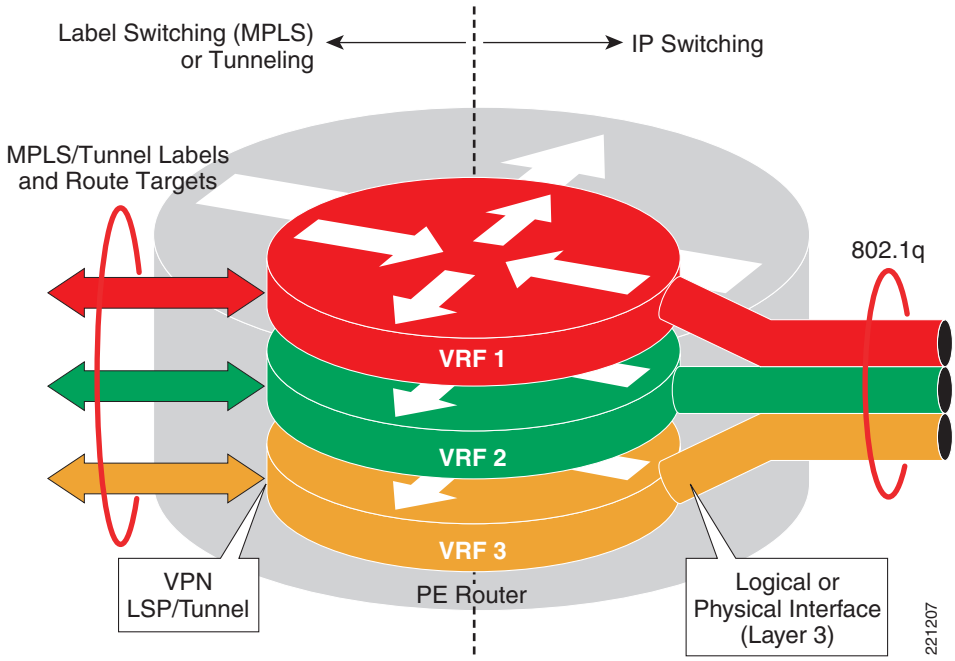
Troubleshooting Tools

Several troubleshooting tools can be used on the PE devices to verify proper connectivity across for each VPN across the MPLS core. See MPLS-Specific Troubleshooting Tools, page 139 for more details on this topic.

Enabling MPLS in the Campus Distribution block

Because the distribution layer devices represent the first L3 hops in the network and where VRFs are first defined, Cisco recommends to position here the PE functionality when deploying MPLS VPN as a path isolation strategy across the campus network. (See Figure 62.)

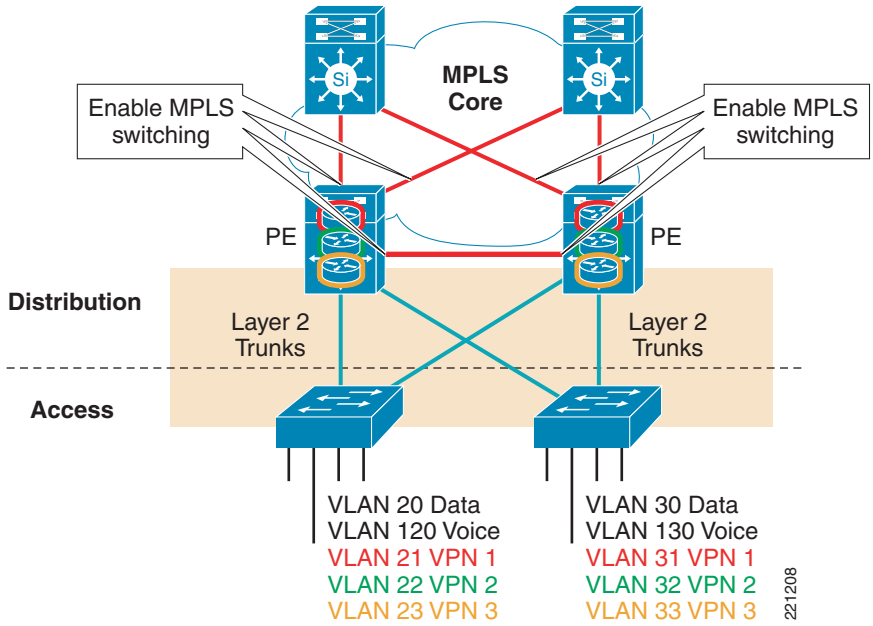
Figure 62 PE Functionality



The Catalyst 6500 platform deployed in the distribution layer needs to have the VRFs defined and the capabilities of communicating on one side with IP switching (toward the access layer devices) and translating that on the other side to MPLS switching (toward the campus core switches). To perform that functionality, the device needs to be able to push VPN labels to the IP packet. This is different from simple VRF-lite support that was for example required when deploying GRE tunnels as path isolation mechanism (see Path Isolation using VRF-Lite and GRE, page 21).

Figure 63 shows an example of enabling MPLS switching.

Figure 63 Enabling MPLS Switching



The configuration required to enable MPLS switching on the interface facing the campus core is as simple as follows:

```
interface TenGigabitEthernet1/1
 description 10GE to core 3
 ip address 10.122.5.31 255.255.255.254
 tag-switching ip
```

Note It is important to note that the actual configuration (retrieved through the **show running-config** command) may show the word “tag-switching” in place of “mpls” on 6500 platforms. This is going to change in future releases of IOS code and it is just a heritage from the past (tag-switching was the pre-standard label switching mechanism supported on Cisco platforms before MPLS was introduced).

LDP Deployment Considerations

After enabling label switching on all the interfaces facing the core, it is also required to enable LDP. LDP is the IETF prescribed way to discover MPLS neighboring devices and transmit label information between the devices. LDP is largely based upon the pre-standard TDP (Tag Distribution Protocol) that was developed by Cisco for tag switching and was later standardized to become MPLS.

When an interface is enabled for label switching (as shown in the previous section), the LDP process starts and tries to discover other MPLS-enabled neighbors (either PE or P devices) by sending LDP hello packets. When a neighbor has been discovered, an LDP session is established with it by setting up a TCP session on the well-known port 646. As a consequence, IP connectivity is required between neighbors to be able to successfully establish the LDP session. After the LDP session has been established, keepalives messages are exchanged between the neighbor devices (by default every 60 seconds), as highlighted in the following output:

```
cr20-6500-1#sh mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 524286
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
TDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

There are several best practices recommendations for deploying LDP in a campus environment, and these are discussed in the following bullet points:

- Configure LDP as label distribution protocol

  As previously mentioned, Cisco originally deployed its own label distribution protocol called Tag Distribution Protocol (TDP). As a consequence of this heritage, Catalyst 6500 platforms use TDP by default on all the MPLS-enabled interface, as follows:

  ```
  cr20-6500-1(config)#mpls label protocol ?
    ldp  Use LDP
    tdp  Use TDP (default)
  ```

  Explicit configuration is then required to change the default behavior and enable the use of LDP:

  ```
  cr20-6500-1(config)#mpls label protocol ldp
  cr20-6500-1(config)#do sh mpls ldp parameters
  Protocol version: 1
  Downstream label generic region: min label: 16; max label: 524286
  Session hold time: 180 sec; keep alive interval: 60 sec
  Discovery hello: holdtime: 15 sec; interval: 5 sec
  Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
  Downstream on Demand max hop count: 255
  LDP for targeted sessions
  LDP initial/maximum backoff: 15/120 sec
  LDP loop detection: off
  ```

- Use loopback interfaces to establish LDP sessions

  Each LDP session between MPLS-enabled neighbors is characterized by an LDP identifier that is use similarly to the OSPF or BGP identifiers. By default, the highest IP address of all defined loopback interfaces is used and if there are no loopbacks, the highest IP address of any other interface is adopted as LDP identifier. The recommendation is to define a specific loopback interface to be used for the establishing of the LDP session. The first reason for doing that is the operational control of the LDP identifier; a second important reason is discussed in the next bullet point. The required configuration is shown as follows:

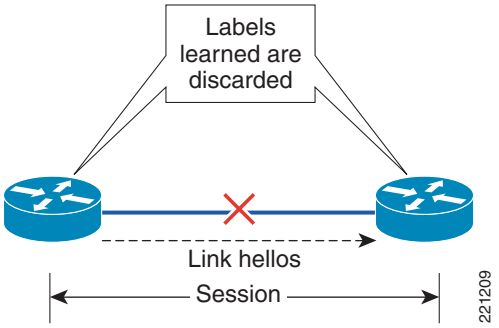
  ```
  interface Loopback10
   description LDP identifier
   ip address 192.168.100.19 255.255.255.255
  end
  !
  mpls ldp router-id Loopback10 force
  ```

Note As discussed before, IP connectivity is required between MPLS-enabled neighbors to establish an LDP session. When defining loopback interfaces to be used as LDP identifiers, it is then critical that the loopback is reachable by adjacent devices. This usually implies that the loopback addresses must be advertised by the IGP running in the network and being thus part of the default global routing table.

- Establish targeted sessions between LDP neighbors

  LDP plays a critical role when discussing convergence in an MPLS-enabled network. As shown in Figure 64, a link failure event between adjacent MPLS-enabled devices causes the failure of the LDP session between them.

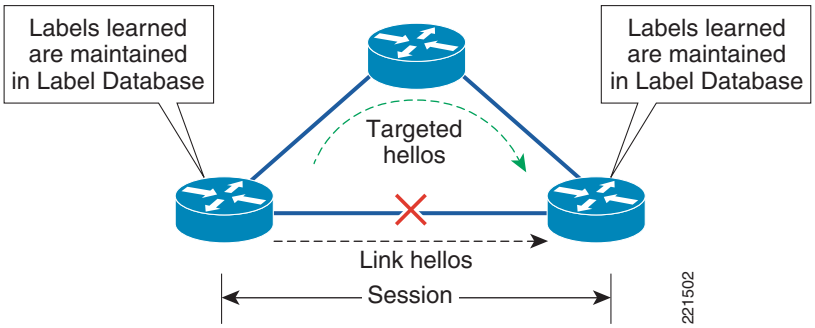
Figure 64 Failure of Regular LDP Session



As highlighted in Figure 64, this means that all the labels that were previously exchanged between the neighbors are now discarded and deleted from the label database. Convergence is usually not an issue for this link failure scenario, because in this case the LDP convergence is almost immediate, and the main factor determining the length of the outage is the time needed by the IGP to converge around the failure.

Different considerations must be made for the reestablishment of the link. Under such a circumstance, the main problem is that IP usually converges much faster than LDP. As a consequence, there may be a temporary incapability to forward label packets until new labels are exchanged and the label database is populated. This does not affect global table traffic (packets can flow also as unlabeled IP data) but it does cause VPN traffic to be dropped (the P device connected to the PE switches traffic) based on the external label. This is usually the IGP label, so if this is missing because of LDP convergence, the switching decision is made based on the actual VPN label, causing the traffic to be dropped or delivered to the wrong destination. A possible workaround for this issue calls for the establishment of targeted sessions between LDP neighbors (see Figure 65.)

Figure 65 Use of LDP Targeted Hellos



As shown in Figure 65, when using targeted hellos between LDP neighbors (for example R1 and R2), the LDP session between these devices is maintained even when the direct link connecting them fails, as long as there is an alternate path for maintaining the TCP session active; in the example, this happens through R3. This means that the MPLS labels that were originally exchanged between the neighbors are kept in the software label database and not discarded; the advantage in doing so is that once the direct link is reestablished, these labels do not need to be learned again, so the IP convergence is the only factor affecting the overall traffic recovery on that link (together with the programming of the hardware label database).

To use this capability, Cisco recommends following three main design recommendations:

- Build a high degree of redundancy when deploying the campus network, so that there is always at least a redundant path connecting each pair of network devices.
- Configure loopback interfaces as LDP identifiers, as previously discussed. In fact, if the LDP session is established by using the IP address of the physical interfaces connecting the neighbor devices, the targeted hellos feature cannot provide any benefit (the TCP session is broken as soon as the physical link fails). Note that it is also required to inject the loopback interface IP addresses into the IGP in use to successfully establish the TCP sessions between neighbors.
- Specify that the LDP session established with the neighbor devices must be a targeted session, as shown in the following configuration sample:

```
cr20-6500-1(config)#mpls ldp neighbor 192.168.100.19 targeted ?
  ldp  Use LDP
  tdp  Use TDP
  <cr>
```

Note It is also optional to specify if LDP or TDP should be used between the LDP neighbors. Cisco recommends to configure the specific label protocol to be used globally, as previously discussed.

From a verification standpoint, as shown in the following example, the LDP session with the neighbor of this example (192.168.100.19) is maintained via the directly connected link (interface Ten1/1):

```
cr20-6500-1#sh ip route 192.168.100.19
Routing entry for 192.168.100.19/32
  Known via "ospf 100", distance 110, metric 2, type intra area
  Last update from 10.122.5.30 on TenGigabitEthernet1/1, 00:00:04 ago
  Routing Descriptor Blocks:
  * 10.122.5.30, from 10.122.5.103, 00:00:04 ago, via TenGigabitEthernet1/1
      Route metric is 2, traffic share count is 1
```

If the physical link fails, the LDP session is maintained in active state via the alternate path (via Ten1/3 and the distribution layer peer):

```
cr20-6500-1(config)#int t1/1
cr20-6500-1(config-if)#shut
cr20-6500-1(config-if)#end
cr20-6500-1#sh ip route 192.168.100.19
Routing entry for 192.168.100.19/32
  Known via "ospf 100", distance 110, metric 4, type inter area
  Last update from 10.137.0.3 on TenGigabitEthernet1/3, 00:00:07 ago
  Routing Descriptor Blocks:
  * 10.137.0.3, from 10.122.5.114, 00:00:07 ago, via TenGigabitEthernet1/3
      Route metric is 4, traffic share count is 1

cr20-6500-1#sh mpls ldp neighbor 192.168.100.19
    Peer LDP Ident: 192.168.100.19:0; Local LDP Ident 192.168.100.5:0
        TCP connection: 192.168.100.19.11094 - 192.168.100.5.646
        State: Oper; Msgs sent/rcvd: 106/85; Downstream
        Up time: 00:15:24
        LDP discovery sources:
          Targeted Hello 192.168.100.5 -> 192.168.100.19, active, passive
        Addresses bound to peer LDP Ident:
          192.168.100.19  172.26.159.146  10.122.5.11     10.122.5.12
          10.122.5.34
        Duplicate Addresses advertised by peer:
          2.2.2.2
```

In addition, note that all the labels learned from the LDP neighbor are still kept in the software label database. Regarding the prefix 10.122.5.2/31 in the example above, it is possible to notice in the following example how tag 23 is still associated to it in the label database. This tag was originally learned from the interface Ten1/1 before of its failure.

```
cr20-6500-1#sh mpls ldp bind neighbor 192.168.100.19
  tib entry: 0.0.0.0/0, rev 154
        remote binding: tsr: 192.168.100.19:0, tag: imp-null
  tib entry: 2.2.2.2/32, rev 2
        remote binding: tsr: 192.168.100.19:0, tag: imp-null
  tib entry: 10.122.5.2/31, rev 72
        remote binding: tsr: 192.168.100.19:0, tag: 23
  tib entry: 10.122.5.6/31, rev 66
        remote binding: tsr: 192.168.100.19:0, tag: 20
  (output suppressed)
```

The hardware label database is instead programmed to use a different label (the LSP is built via the alternate path now that Ten1/1 has failed); this can be verified by looking at the specific label that is used to reach one of the prefixes shown above (**10.122.5.2** in this example):

```
cr20-6500-1#sh mpls forwarding-table 10.122.5.2
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
52     21          10.122.5.2/31     0          Te1/3      10.137.0.3
```

As expected, the outgoing label currently in use is 21 out of interface Ten1/3 (and not the tag 23 that was originally learned via Ten1/1). However, as soon as the link is recovered, the hardware is reprogrammed with the updated information without requiring a new learning of that label:

```
cr20-6500-1(config)#int t1/1
cr20-6500-1(config-if)#no shut
cr20-6500-1#sh mpls forwarding-table 10.122.5.2
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
52     23          10.122.5.2/31     0          Te1/1      10.122.5.30
```

In summary, the use of loopback interfaces for establishing LDP-targeted sessions between neighbor network devices provides for fast hardware recovery for failed links and thus represents the recommended best practice. In addition, the use of loopback interfaces addressed from a specific and well identifiable IP pool provides a further advantage that is discussed in Tagging or not-Tagging Global Table Traffic, page 127.

MP-iBGP Deployment Considerations

In an MPLS VPN design, the exchange of VPN routes is achieved by using an additional control plane element called Multi-Protocol BGP (MP-BGP), which is an extension of the existing BGP-4 protocol. In the context of this guide, MP-BGP is introduced only as an overlay protocol to provide the capabilities for exchanging VPN routes. Very large networks can be deployed as separate autonomous systems (AS), and in such scenarios, the use of BGP may be required also to connect these separate AS and exchange global table routes. The recommended design discussed here is instead constituted by a single AS and an IGP deployed end-to-end, so that there is no requirement for BGP in global table.

As a consequence, MP-BGP needs to be configured only between the PE devices, because they are the only ones containing VPN routes in the various VRF routing tables. A direct consequence of the fact that the main MPLS VPN strength is to provide any-to-any connectivity inside each defined VPN is the requirement for the PE devices to establish MP-iBGP connections between them in a fully-meshed fashion. By deploying route reflectors, it is possible to relax this requirement, thus improving the scalability of the overall solution.

MP-iBGP is required within the MPLS VPN architecture because the BGP updates exchanged between PE devices need to carry more information than just an IPv4 address. At a high level, the following three pieces of information are critical to the MPLS VPN functionality and that are exchanged through MP-iBGP:

- VPNv4 addresses—Address prefixes defined in the context of each VPN that need to be communicated between the various PE devices to provide connectivity inside each VPN. A VPNv4 address is achieved by concatenating together the IPv4 prefix and a 64-bit entity called a route distinguisher (RD). A unique RD needs to be used for each VRF defined on the PE device. The RD uniqueness contributes to the uniqueness of each VPNv4 prefix, allowing the support of overlapping IPv4 prefixes between separate VPNs.
- MPLS VPN label information—Each PE allocates a specific MPLS label for each defined VPN prefix. This is the more internal label that is pushed in each MPLS packet before sending it to the MPLS core, and is used by the receiving PE to determine in which VPN to route the packet.
- Extended BGP communities—The most important of these extended communities is called the route target and represents a 64-bit value that is attached to each BGP route. The value of the route target determines how the VPN routes are exported and imported into each VPN. Basically, every VPNv4 routes received by a PE may have one of more route target associated to it; depending on the route targets configured locally on the receiving PE for each VRF, the route is either imported or ignored for that specific VRF. Using route targets provides great flexibility to provision many different VPN topologies. In the context of this guide, how to provide any-to-any connectivity inside each VPN is discussed. For an explanation of how to deploy a hub-and-spoke topology as opposed to an any-to-any topology, see the *Network Virtualization—Services Edge Design Guide* (OL-13637-01).

When discussing the deployment of MPLS VPN in a campus environment, the following specific recommendations should be followed:

- Differing from a traditional service provider environment, the first thing to consider when deploying MPLS VPN in a campus distribution block is the absence of a traditional CE device, because all the VPN subnets are directly connected to the PE devices deployed at the distribution layer (given that the access layer switches are functioning as L2 devices). This means that there is no need for a

CE-PE control protocol. All the VPN subnets are showing into each defined VRF as directly connected, which essentially allows injecting them into MP-BGP by simply configuring the **redistribute connected** option, as follows:

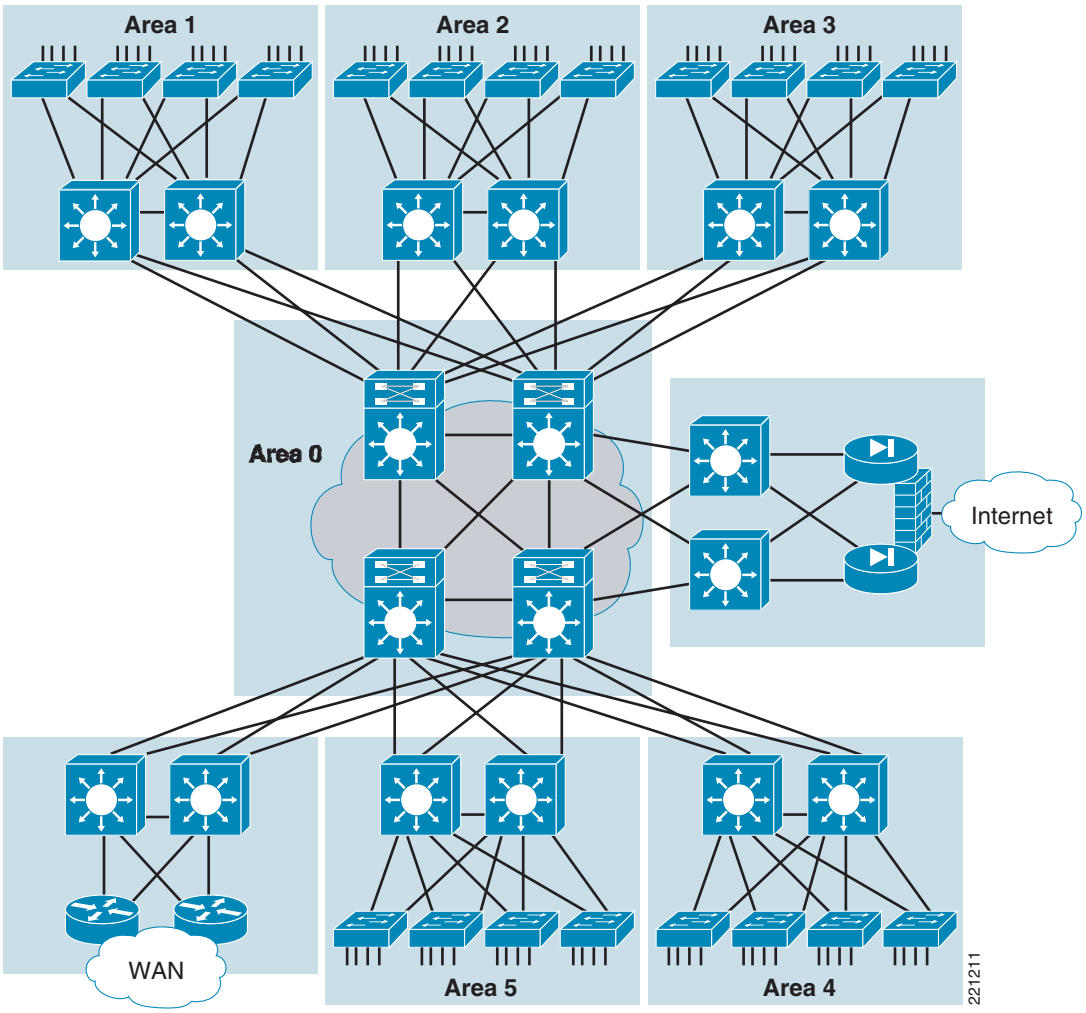
```
router bgp 64000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf v1
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
```

- MP-iBGP sessions should be established by using loopback interfaces. This brings the obvious advantage of allowing the iBGP session to remain active as long as there is an available path connecting to the loopback IP address. In addition, there are also some operational advantages in assigning an IP address to the loopback interfaces taken from a unique and easy identifiable subnet. This point is discussed in LDP Deployment Considerations, page 103; it is in fact recommended to use the same loopback interface as the LDP identifier and for establishing MP-iBGP sessions. Another reason for doing this is discussed in Tagging or not-Tagging Global Table Traffic, page 127. When using loopback interfaces, the configuration look like the following sample:

```
interface Loopback10
 description mBGP anchor point
 ip address 192.168.100.5 255.255.255.255
!
router bgp 64000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.100.1 remote-as 64000
 neighbor 192.168.100.1 update-source Loopback10
```

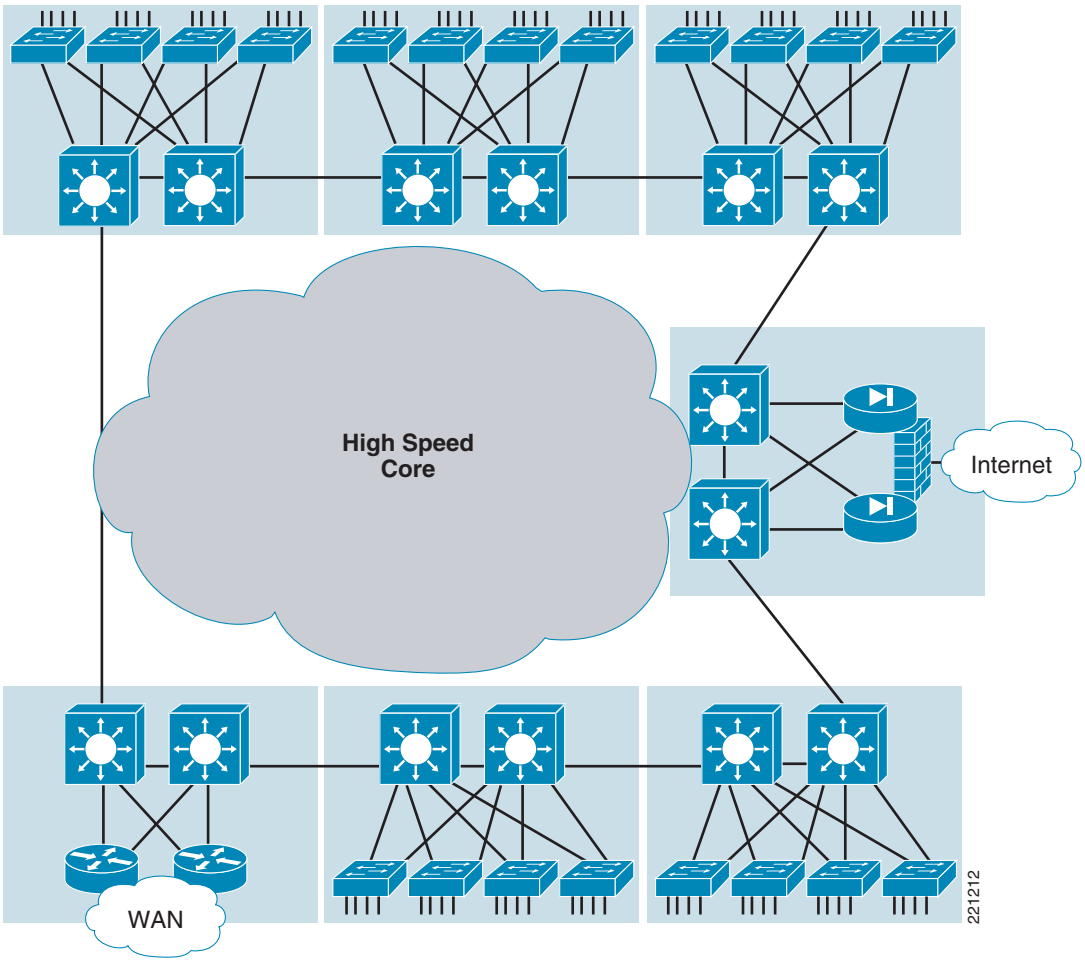
Special considerations need to be made about the loopback interfaces when using OSPF as the IGP in the global table. Following the campus design recommendation, each distribution block is usually deployed as a separate OSPF area, with the core connecting all the blocks together as area 0. As shown in Figure 66, the transit link connecting the distribution layer devices is configured as part of the specific OSPF area defined in each distribution block.

Figure 66 OSPF Area Definition in Campus Networks



A special consideration on this regard must be done for topologies connecting different distribution blocks in a ring, as shown in Figure 67.

Figure 67 Campus Ring Topology



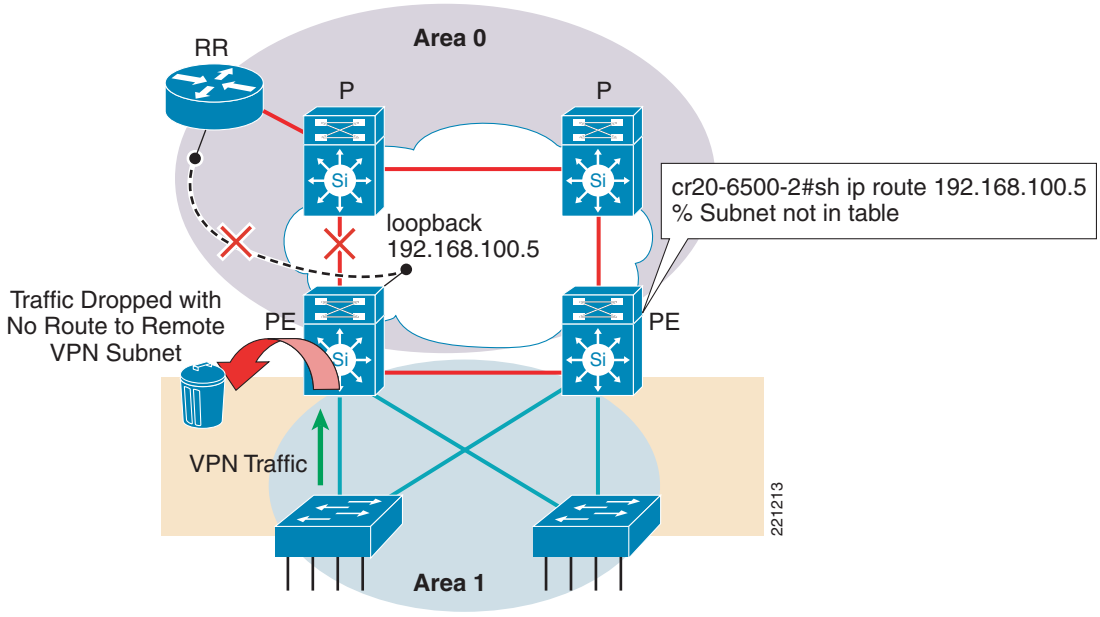
When deploying OSPF in this topology, the links connecting the various distribution blocks between them are positioned in area 0. Following the traditional recommendation of configuring the transit links between distribution layer peers as part of the specific OSPF area defined in each specific block thus creates a fragmented area 0. The best practice recommendation to solve this problem is to add another link (physical or logical) between the distribution layer peers configured in area 0. Doing this allows maintaining the summarization of routes toward the core, with all the advantages in terms of routing scalability, management, convergence, and stability.

Concerning the loopback interfaces, the recommendation is to keep them as part of the area as well, as shown in the following configuration sample:

```
interface Loopback10
 description mBGP anchor point
 ip address 192.168.1.3 255.255.255.255
!
router ospf 100
 router-id 10.122.5.115
 log-adjacency-changes
 timers throttle spf 10 100 5000
 timers throttle lsa all 10 100 5000
 timers lsa arrival 80
 network 192.168.1.3 0.0.0.0 area 1
```

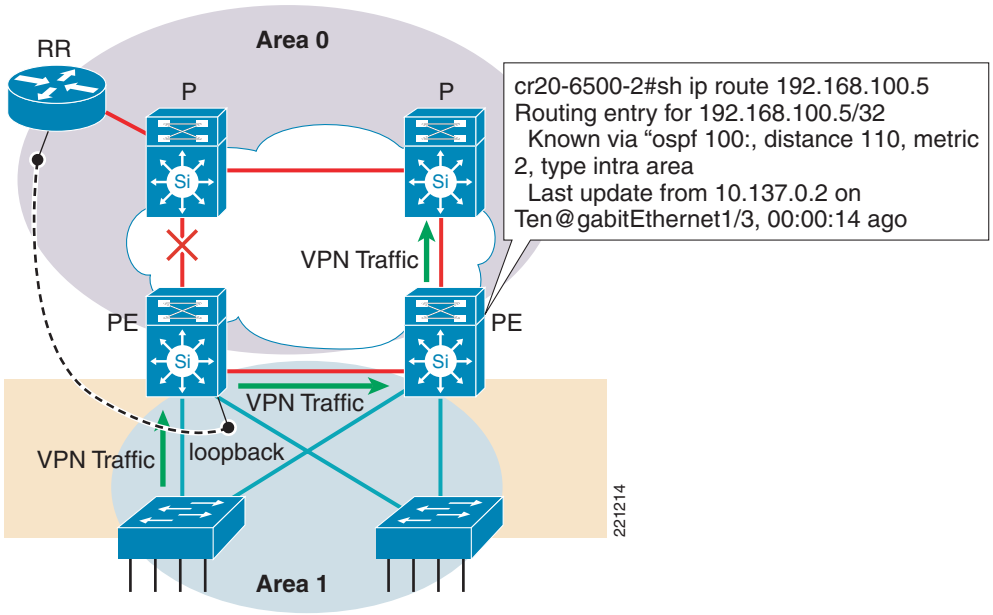
The scenario where this recommendation becomes mandatory is the partial mesh topology in case the distribution layer device functioning as HSRP active loses its connectivity to the core. In this case, if the loopback used for MP-iBGP peering was deployed in area 0, the iBGP session between the PE and the route reflector fail because the RR no longer have a valid path toward the PE loopback. This is because the peer PE device is not learning the loopback information via the transit link, because the loopback is in area 0 but the transit link is in area 1. However, the PE on the left still remains the HSRP active, so all the traffic originated from local VPN subnets is sent up to this device when destined to remote VPN subnet and is dropped causing a network black hole, as shown in Figure 68.

Figure 68 Assigning Loopback to Area 0



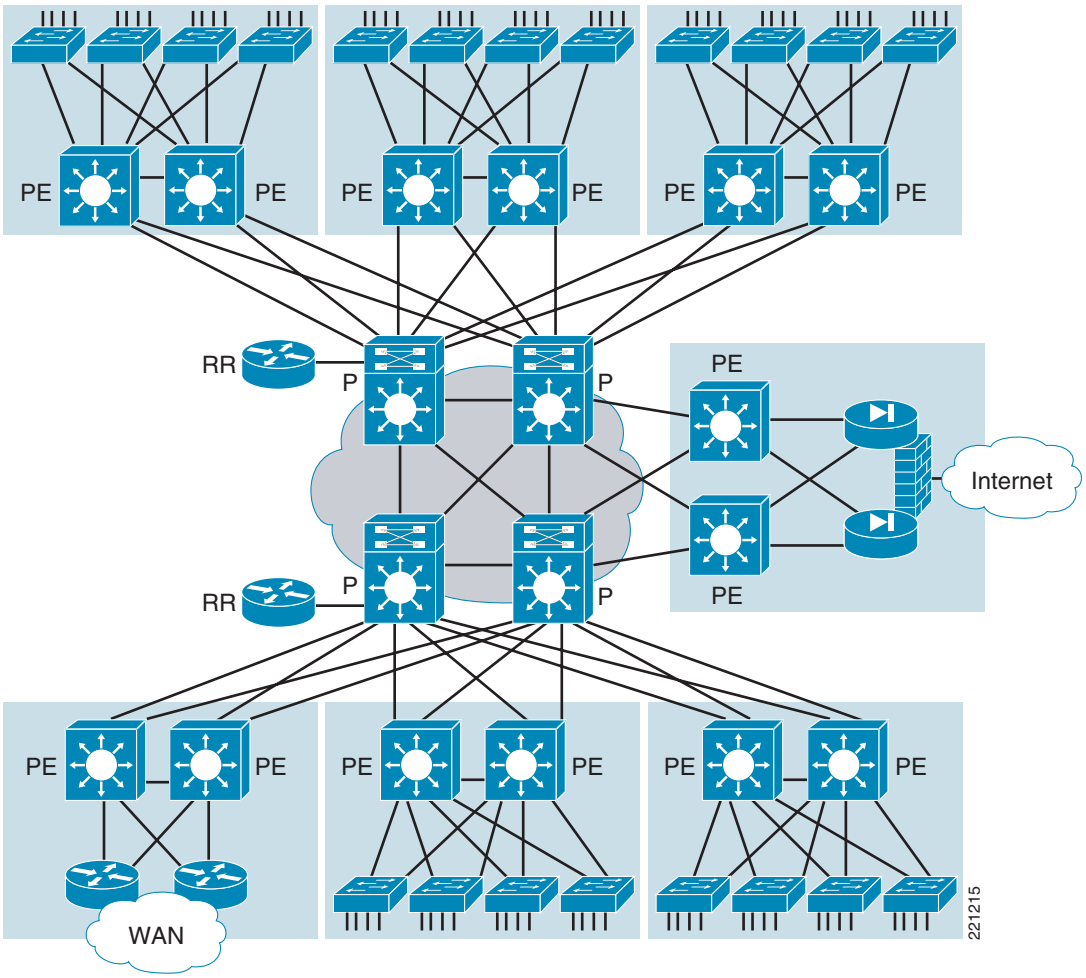
The recommended best practice to solve this issue is to deploy the loopback as part of the specific OSPF area defined in the distribution block. This allows the peer PE to learn the loopback information as an intra-area route via the transit link. The MP-iBGP session between the RR and the left PE then remains active and the upstream VPN traffic is therefore sent over the transit link between PEs, as shown in Figure 69.

Figure 69 Loopback Not Part of OSPF Area 0



- Given the fact that MP-iBGP sessions need to be established between all the PE devices defined in the network, Cisco recommends using route reflectors for a better scalability and manageability of the solution. The route reflector should be deployed on standalone devices connected, for example, to the P core devices, as shown in Figure 70.

Figure 70 Positioning of Route Reflectors



One of the main advantages in using standalone devices is stability. Upgrade of code to P or PE devices can be performed without touching the RR that can continue performing its function. Also, the MP-BGP configuration required on each PE devices becomes identical; all the PEs have to peer with the two route reflectors, as shown in the following example. This design recommendation considerably reduces maintenance time and improves operational ease of troubleshooting.

```
router bgp 64000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.100.1 remote-as 64000
 neighbor 192.168.100.1 update-source Loopback10
 neighbor 192.168.100.2 remote-as 64000
 neighbor 192.168.100.2 update-source Loopback10
 !
 address-family vpnv4
 neighbor 192.168.100.1 activate
 neighbor 192.168.100.1 send-community extended
 neighbor 192.168.100.2 activate
 neighbor 192.168.100.2 send-community extended
 exit-address-family
```

On the RR side, the configuration is straightforward:

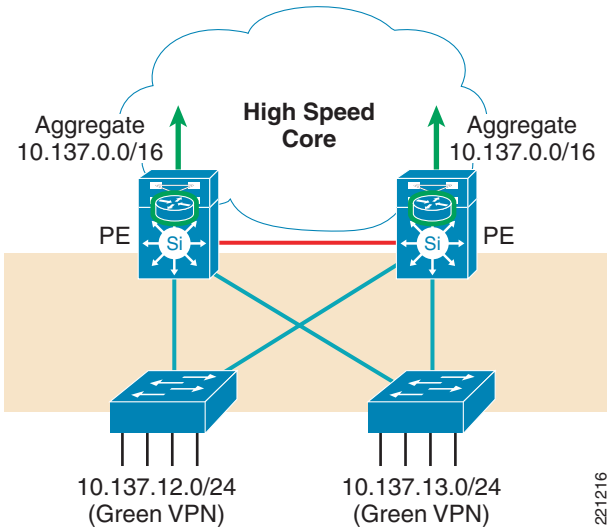
```
router bgp 64000
```

```
 no bgp default ipv4-unicast
 neighbor RR-clients peer-group
 neighbor RR-clients remote-as 64000
 neighbor RR-clients update-source Loopback10
 neighbor 192.168.100.3 peer-group RR-clients
 neighbor 192.168.100.4 peer-group RR-clients
 neighbor 192.168.100.5 peer-group RR-clients
 neighbor 192.168.100.6 peer-group RR-clients
!
address-family vpnv4
 neighbor RR-clients activate
 neighbor RR-clients send-community extended
 neighbor RR-clients route-reflector-client
 neighbor 192.168.100.3 peer-group RR-clients
 neighbor 192.168.100.4 peer-group RR-clients
 neighbor 192.168.100.5 peer-group RR-clients
 neighbor 192.168.100.6 peer-group RR-clients
 exit-address-family
```

Note When positioning the RRs as separate network devices (as in the recommended model displayed in figure above), no MPLS or VRF definitions are required on these devices.

- Aggregation of VPN subnets—Summarization of VPN routes from each campus distribution block toward the core is not recommended best practice because it may lead to a black hole situation under a specific failure scenario. As shown in Figure 71, assume that both PEs belonging to the distribution block are aggregating VPN routes toward the core; for example, advertising a /16 super-net.

Figure 71 Summarizing VPN Routes



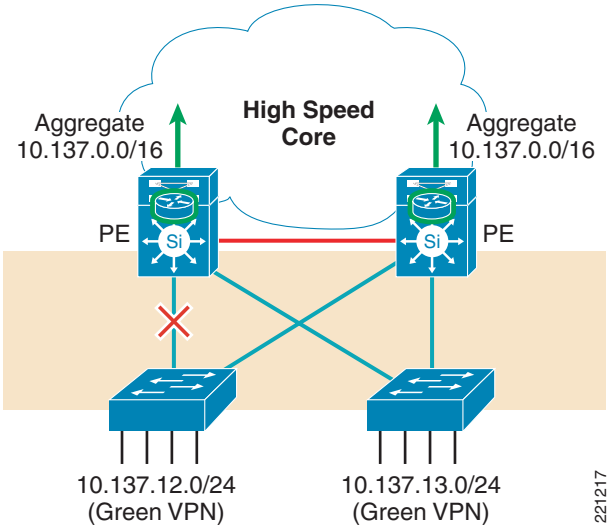
A look in the VRF routing table of each PE shows the VPN subnet directly connected and the summary pointing to Null0:

```
cr20-6500-1#show ip route vrf v1
Routing Table: v1
Codes: 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, E - EGP
       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
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
B       10.137.0.0/16 [200/0] via 0.0.0.0, 00:43:59, Null0
C       10.137.13.0/24 is directly connected, Vlan13
C       10.137.12.0/24 is directly connected, Vlan12
```

Now assume one of the uplink from the access layer to the distribution switch fails, as shown in Figure 72.

Figure 72 Link Failure when Summarizing VPN Routes



Without VPN route aggregation, the PE on the left directly connected to the failed link learns (via BGP) the path toward the subnet 10.137.12.0 via the peer PE device. When summarizing instead, the PE ignores the summary learned from the peer because it already has a summary route pointing to Null0, as shown in the following example:

```
cr20-6500-1#show ip route vrf v1
Routing Table: v1
Codes: 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, E - EGP
       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
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
B       10.137.0.0/16 [200/0] via 0.0.0.0, 00:43:59, Null0
C       10.137.13.0/24 is directly connected, Vlan13
```

As a consequence, the PE starts dropping all the traffic delivered to it from the core of the network and destined to the specific 10.137.12.0 subnet. This is the reason why summarization of VPN routes from each distribution block is not a recommended best practice.

Configuring the Core Devices (P Routers)

The configuration of the devices building the core of the MPLS network (P devices) is much simpler than the one discussed in the previous section for PE switches because of the following two main reasons:

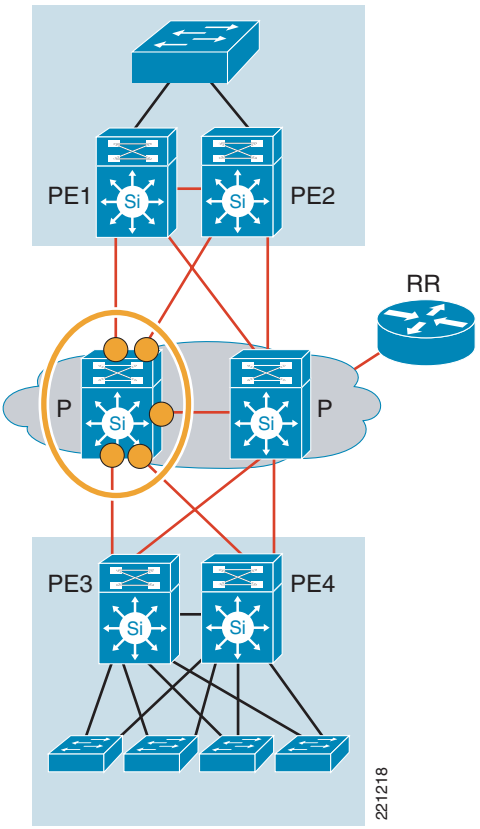
- P devices do not generally require any VRF configuration or network services virtualization. These functionalities are deployed only on the PE switches sitting at the edge of the MPLS network. The main task of the P switches consists in label switching the received packets, allowing for the establishment of LSPs across the network infrastructure (it has already been discussed how these LSPs can be used to switch both global table and VPN traffic).
- As a direct consequence of the previous point, there is no the requirement for the additional control plane protocol MP-BGP to be deployed on P devices. The only routing protocol in use is the IGP traditionally deployed to establish global table connectivity.

Note The requirements for deploying P devices in the core of the network are the same as PE switches. Therefore, only Catalyst 6500 switches with Supervisors equipped with PFC3B or higher are currently available for this role.

Given the considerations above, the following are the basic configuration steps required for P (core) switches deployment. As previously mentioned, the assumption is that global table configuration (routing, IP addressing, and so on), is already in place before starting the virtualization of the network infrastructure.

Step 1 Enable MPLS switching on all the physical interfaces connecting the P devices to other P or PE switches, as shown in Figure 73.

Figure 73 Enabling MPLS on P Devices




```
interface TenGigabitEthernet1/1
 description 10GE to PE3
 ip address 10.122.5.37 255.255.255.254
 tag-switching ip
```

Step 2 Configure LDP parameters similarly to PE:

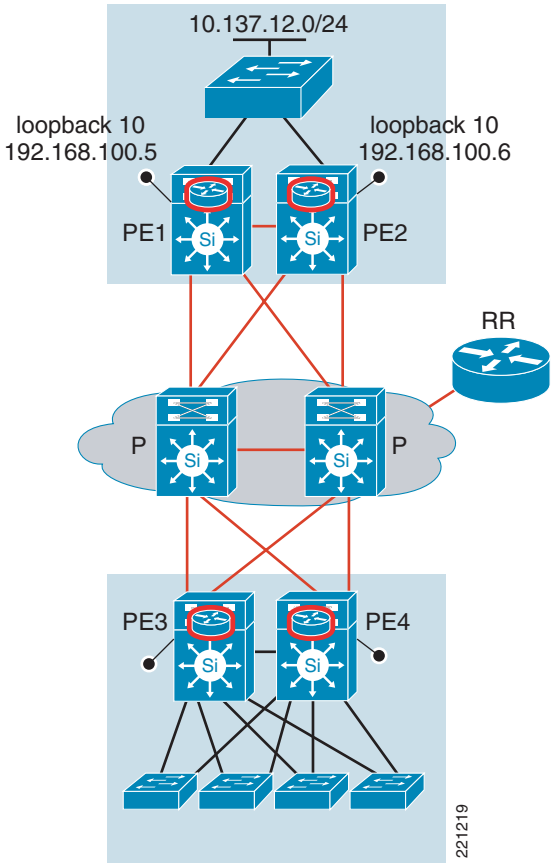
- Explicitly enable standard LDP
- Define a loopback interface to be used as the LDP identifier
- Inject the loopback /32 address into IGP
- Establish targeted sessions between LDP neighbors

Redundancy and Traffic Load Balancing

Because of the business-critical functions usually supported by campus networks, the design has evolved to one supporting a high degree of redundancy to achieve the required high availability. This leads to the deployment of redundant devices in the core and distribution layers, redundant supervisors in the access layer, and redundant links connecting the various layers of the hierarchical network. The application traffic in the VPNs is also considered mission-critical and needs to be protected in a similar fashion as the global table traffic. Therefore, it is important to understand how to use the infrastructure redundancy also for that purpose.

To achieve this, several configuration steps need to be implemented. To understand this point better, see the network diagram in Figure 74.

Figure 74 Achieving Redundancy and Traffic Load Balancing

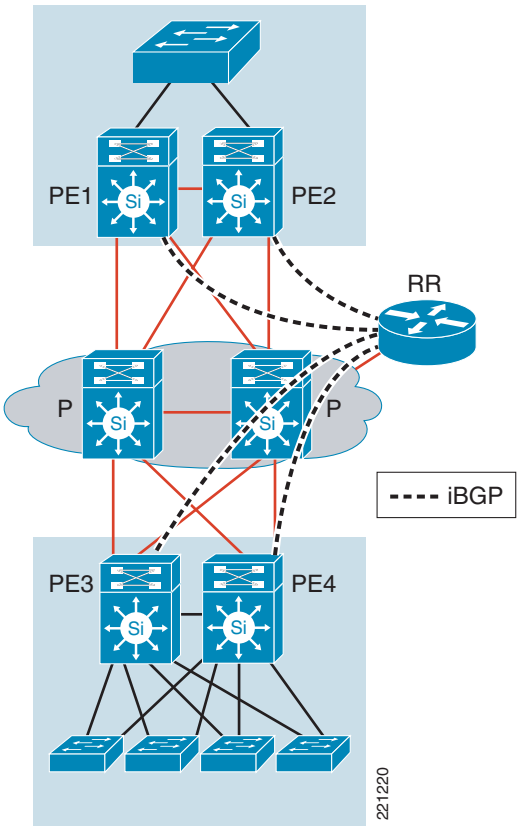


In Figure 74, PE1 and PE2 are connected to a subnet (10.137.12.0/24) mapped to VRF v1 (thus part of a specific VPN). Because all the devices in the example are connected in a fully meshed fashion, it is desirable for the VPN traffic flowing between the two distribution blocks to also benefit from this link redundancy.

For this to happen, the first design recommendation is to configure a different RD value for the two PE devices belonging to the same distribution block. To understand the reasons for this choice, a brief review of how the PE devices on the bottom receive the VPN routes from the upper PEs is useful.

As shown in Figure 75, when deploying RRs, all the PE devices must establish an MP-iBGP session with the RRs (for simplicity sake, only one RR is discussed in this example).

Figure 75 Establishing MP-iBGP Sessions with Route Reflector



PE1 and PE2 must advertise the same IPv4 subnet (10.137.12.0/24) to the RR via MP-IBGP. By default, the RR chooses one of the two VPNv4 updates received and "reflects" the best one to the other RR clients; in this example, the bottom PE3 and PE4. As a consequence, if the RD value configured on PE1 and PE2 is the same, they both advertise the same VPNv4 route to the RR, and the RR reflects only the better one to the bottom PEs. Configuring a distinct RD value instead has the consequence of making the VPNv4 update unique sent by PE1 and PE2 for the same IPv4 prefix 10.137.12.0. The RR then "reflects" both VPNv4 prefixes to the bottom PEs.

The configuration required for achieving load balancing and redundancy is therefore the following:

- PE1

```
ip vrf v1
 rd 64001:1
 route-target export 64000:1
 route-target import 64000:1
```

- PE2

```
ip vrf v1
 rd 64002:1
 route-target export 64000:1
 route-target import 64000:1
```

Notice how the route-target values need to remain the same on both PEs because they both need to import into the specific VPN routing table the same updates received by remote PEs. This is required on all the PEs when the goal is to achieve any-to-any connectivity inside each VPN.

At this point, the bottoms PEs receive two separate VPNv4 updates for the same IPv4 prefix 10.137.12.0/24. However, an additional configuration step is still required for them to import both the routes in the VPN routing table. This is because by default, the BGP process on the receiving PE devices installs only the best route in the routing table. To change this behavior, the following additional configuration step is required:

- PE3/PE4

```
router bgp 64000
!
address-family ipv4 vrf v1
 maximum-paths ibgp  2 import 2
```

After configuring the above command, the BGP process on the bottom PEs installs both routes received from the upper PEs in routing table, and these routes are consequently imported into the control plane relative to VRF v1, as follows:

```
PE3#sh ip route vrf v1 10.137.12.0
Routing entry for 10.137.12.0/24
  Known via "bgp 64000", distance 200, metric 0, type internal
  Last update from 192.168.100.6 2w3d ago
  Routing Descriptor Blocks:
    * 192.168.100.6 (Default-IP-Routing-Table), from 192.168.100.2, 2w3d ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
    192.168.100.5 (Default-IP-Routing-Table), from 192.168.100.1, 2w3d ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Note This happens only with equal cost routes. It is possible also to import unequal cost routes with the command **maximum-paths ibgp unequal-cost**.

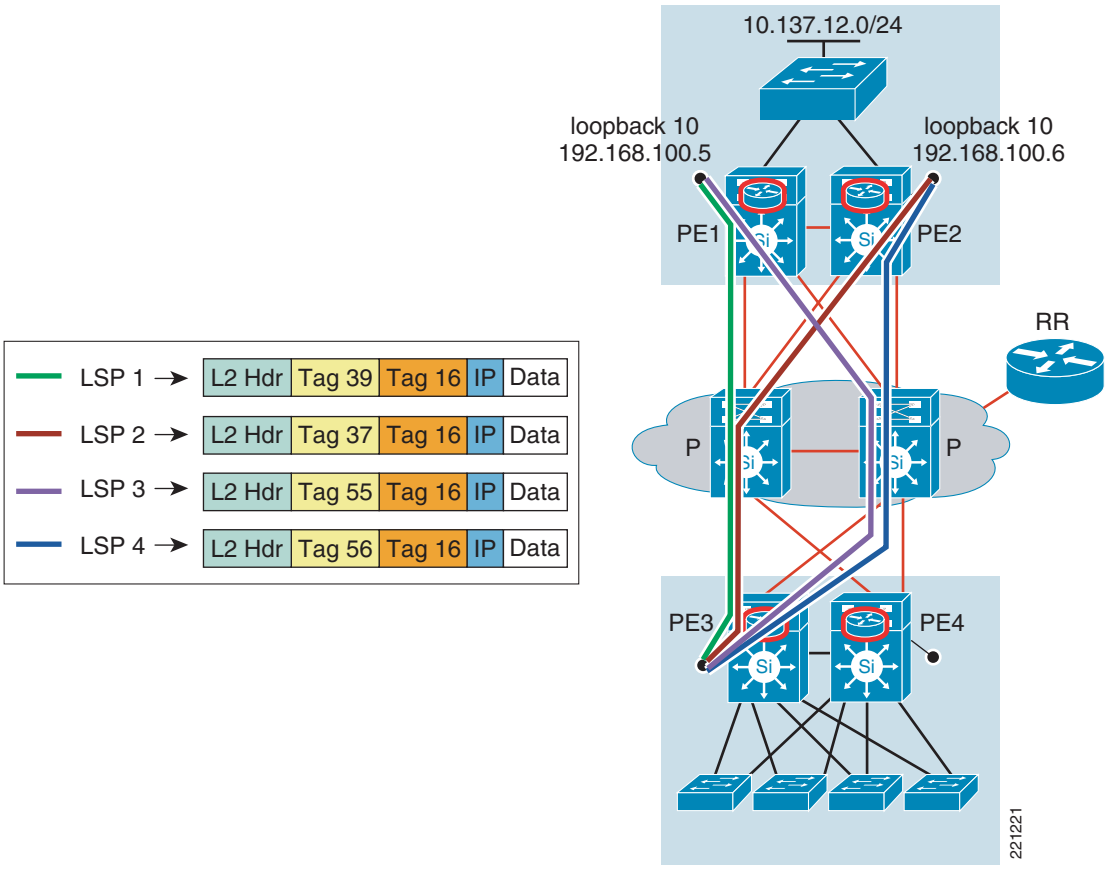
Now that load balancing is achieved from the point of view of the control plane, the discussion needs to focus on how the traffic is actually sent over the physical link; that is, how load balancing is obtained from a data plane point of view on Catalyst 6500 platforms.

As shown in the network diagram above, in a fully meshed design each PE has a redundant equal cost path that can be used to reach the loopback interfaces of the remote PEs. Because each VPN route is then learned from both PEs, the consequence is that each PE is able to send VPN traffic over four distinct Label Switched Paths (LSPs), two on each physical link connecting the PE device to the core. This can be verified as follows:

```
PE3#sh mls cef vrf v1 10.137.12.0
Codes: decap - Decapsulation, + - Push Label
Index  Prefix              Adjacency
3219   10.137.12.0/24      Gi1/3           16(+),56(+) (Hash: 0001)
                           Gi1/2           16(+),39(+) (Hash: 0002)
                           Gi1/3           16(+),55(+) (Hash: 0004)
                           Gi1/2           16(+),37(+) (Hash: 0008)
```

As shown in Figure 76, the same inner MPLS VPN label 16 is used to send traffic toward the destination subnet, whereas a different outer label is inserted to label switch traffic to the remote PEs.

Figure 76 Establishment of Redundant LSPs



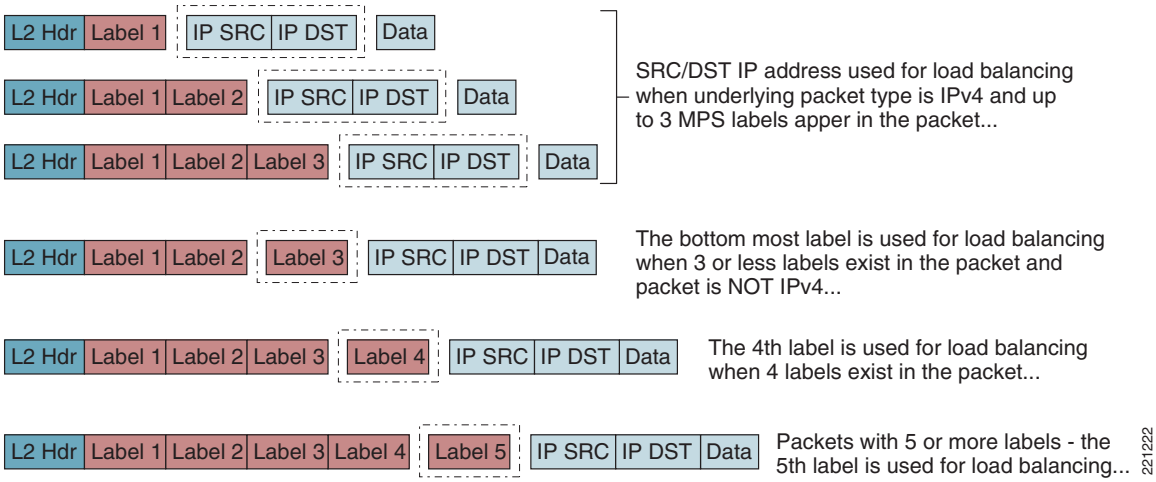
Imposing these labels allows each PE to build the four distinct LSPs to reach the remote PE loopback interfaces (192.168.100.5 and 192.168.100.6). This can be verified as follows:

```
Bottom_PE_Left#sh mls cef 192.168.100.5
Codes: decap - Decapsulation, + - Push Label
Index  Prefix              Adjacency
82     192.168.100.5/32    Gi1/3           55(+) (Hash: 0001)
                           Gi1/2           37(+) (Hash: 0002)
Bottom_PE_Left#sh mls cef 192.168.100.6
Codes: decap - Decapsulation, + - Push Label
Index  Prefix              Adjacency
84     192.168.100.6/32    Gi1/3           56(+) (Hash: 0001)
                           Gi1/2           39(+) (Hash: 0002)
```

Note that two LSPs are formed to reach the loopback interfaces of each remote PE. These LSPs are built out of the two physical interfaces connecting the PE devices to the core.

Now the question is how the PE decides which LSP to use for each specific packet. To answer this, keep in mind how the Catalyst 6500 platforms behave for MPLS traffic in the presence of redundant equal cost paths. Figure 77 describes the various possible scenarios.

Figure 77 MPLS Load Balancing on Catalyst 6500



Because only two labels are imposed on each packet when switching MPLS VPN traffic, the consequence is that the first option is valid in that case. This means that packets are assigned to each LSP based on the source and destination IP addresses pair; therefore, per-flow LSP assignment is performed. This can be easily verified with the following commands:

```
cr23-6500-1#sh mls cef exact-route vrf v1 10.138.12.11 10.137.12.11
Interface: Gi1/3, Next Hop: 224.0.6.84, Vlan: 1020, Destination Mac: 0009.448f.8200

cr23-6500-1#sh mls cef exact-route vrf v1 10.138.12.11 10.137.12.12
Interface: Gi1/3, Next Hop: 224.0.6.86, Vlan: 1020, Destination Mac: 0009.448f.8200

cr23-6500-1#sh mls cef exact-route vrf v1 10.138.12.11 10.137.12.13
Interface: Gi1/2, Next Hop: 224.0.6.87, Vlan: 1019, Destination Mac: 0005.3142.c400

cr23-6500-1#sh mls cef exact-route vrf v1 10.138.12.11 10.137.12.15
Interface: Gi1/2, Next Hop: 224.0.6.85, Vlan: 1019, Destination Mac: 0005.3142.c400
```

Changing the destination IP address (and thus the flow), a different physical interface and corresponding next-hop value is used. The combination physical interface/next-hop MAC address identifies a different LSP in each case.

It is important to note that using distinct RDs on the two PE devices belonging to the same distribution block causes a larger utilization of memory resources on the PE itself. To understand the reason, it is required to analyze the logic behind the use of RDs on the PE devices. Every time a PE receives a new VPNv4 route (from the route reflector in this specific design), it does the following:

- If the RD of the received route is equal to the RD locally defined on the PE for that specific VRF, the route is imported in the BGP table (assuming also that the route target is configured to allow this).
- If the RD of the received route is different from the local RD, the PE imports the route in the BGP table (under the "section" corresponding to the locally defined RD), and it also keeps a copy in a different section of the BGP table corresponding to the received RD value.

Note This logic was deployed essentially to allow to keep track of which PE devices sent each route, under the assumption that each PE defines a unique RD for the same VRF (this is typical for example in a service provider environment).

Still referring to the example discussed above, because the values used for the pair of PEs are common between all the various distribution blocks (but unique between the PEs deployed in the same block), the VPNv4 routes received, for example, by the PEs in the upper distribution block from the PEs in the lower distribution block would be characterized by two distinct RD values. By looking at the BGP table on each of these PE, the increase of memory required to store this information is evident, as follows:

```
cr20-6500-1#sh ip bgp vpnv4 all
BGP table version is 11740, local router ID is 192.168.100.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 64001:1 (default for vrf v1)
* i10.138.12.0/24   192.168.100.10           0    100      0 ?
* i                 192.168.100.10           0    100      0 ?
* i                 192.168.100.9            0    100      0 ?
*>i                 192.168.100.9            0    100      0 ?
<SNIP>
Route Distinguisher: 64002:1
* i10.138.12.0/24   192.168.100.10           0    100      0 ?
*>i                 192.168.100.10           0    100      0 ?
```

As shown above, the route 10.138.12.0 in the section for default RD for the VRF v1 is imported as learned by both remote PEs (192.168.100.9 and 192.168.100.10). However, in the section for RD 64002:1, it is imported as learned only by one of the two remote PEs (the one configured with that specific RD in VRF v1). If all the PEs used the same RD, the second part of the information would not be present, thus saving memory. At the same time, all the characteristics of load balancing and redundancy discussed in this section would not be achieved. It is also worth considering that using separate RD values on each PE defined in the campus network still allows load balancing, but causes excessive memory use to store all the routes received from the other PEs with unique RDs on each PE. As a consequence, the recommended best practice is to have unique RDs between the two PEs belonging to the same campus distribution block, but reusing these values for all the pairs of PE deployed in the other distribution blocks.

Dealing with MTU Size Issues

Every time a tunneling technology is deployed, concerns about MTU size usually arise. Configuring MPLS VPN causes two additional tags to be imposed on each IP packet. This causes an increase of up to 8 bytes to the overall IP size of the packet. Assuming that the endpoints are generating IP packets with full 1500-byte sizes, it is logical to expect some problems to arise. The issues generally arise when the 1500-byte packets reach the PE devices that are responsible for MPLS label imposition.

If the DF bit in the IP packet is set to 1 (this is generally the case because the endpoint sets the bit to perform path MTU discovery), the PE is not able to add the 8 bytes and then send a packet out of the interfaces connected to the core, assuming they are configured with the default 1500-byte MTU size. At the same time, the PE is not able to fragment the packet because of the DF bit setting, so it drops the packet and returns an Destination Unreachable ICMP message to the source of this IP datagram, with the code indicating fragmentation needed and DF set (type 3, code 4). When the source station receives the ICMP message, it lowers the send message segment size (MSS), and when TCP retransmits the segment, it uses the smaller segment size.

This process works assuming that the end station is actually receiving the ICMP message and it is able to properly lower the MTU size of the generated IP packets. If either of these conditions are not met (for example, because the endpoint is not able to properly process the ICMP message and consequently lower the MTU of the packets it generates), the end stations continue to send full-size IP packets and the PE device keeps dropping them, effectively blackholing all the VPN traffic. For this reason, a different mechanism should be deployed to ensure that the VPN traffic is never dropped because of MTU-related

issues. This also helps with UDP traffic characterized by large frame sizes (as for example, the one generated by video applications) for which the Path MTU discovery mechanism cannot be applied anyway.

The following two solutions to this problem can be deployed:

- Configure jumbo-frame support on all the MPLS enabled interfaces

  The first method consists in increasing the MTU of the physical interfaces enabled for label switching. Because the default MTU size supported on Ethernet interfaces is 1500 bytes, increasing that value to at least 1508 allows the successful transmission of the MPLS labeled packets, both for the global table and VPN traffic. The required configuration is as follows:

```
interface TenGigabitEthernet1/1
mtu 1508
tag-switching ip
```

Note Jumbo frame support must be configured on all the MPLS-enabled interfaces of P and PE devices.

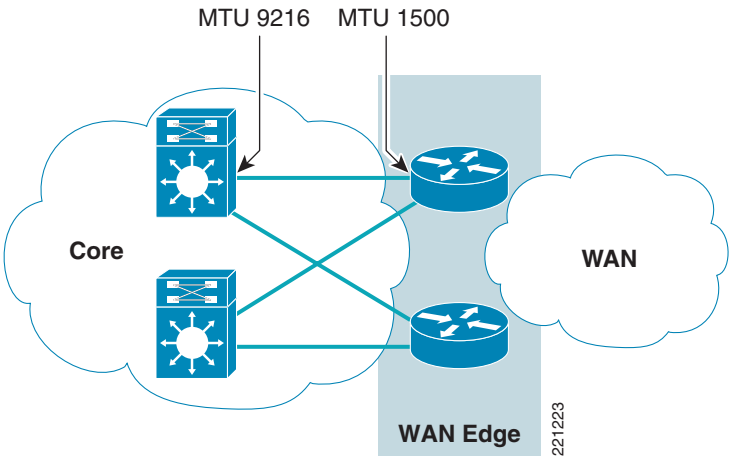
The Catalyst 6500 platform can support jumbo frame sizes as of release 12.1(1)E for Native IOS. However, this support is dependent on the type of line cards that you use. There are generally no restrictions to enable the jumbo frame size feature. You can use this feature with trunking/non-trunking and channeling/non-channeling. As shown in the configuration sample above, a value of 1508 is enough to account for the two MPLS labels added for VPN traffic. However, the maximum jumbo frame size supported on the individual port is 9216; an application specific integrated circuit (ASIC) limitation limits the MTU size to 8092 bytes on the following 10/100-based line cards:

- WS-X6248-RJ-45
- WS-X6248A-RJ-45
- WS-X6248-TEL
- WS-X6248A-TEL
- WS-X6348-RJ-45
- WS-X6348-RJ-45V
- WS-X6348-RJ-21

Note The WS-X6516-GE-TX is also affected at 100 Mbps; whereas at 10/1000 Mbps, up to 9216 bytes can be supported.

One specific issue may arise when modifying the MTU size of the physical interface, which is related to the fact that OSPF does not allow the establishment of adjacencies between devices that have configured a different MTU size on their connecting interfaces. For example, this can be the case when connecting the WAN edge devices to the campus core, as shown in Figure 78.

Figure 78 MTU Mismatch



It may well happen that the network device deployed in the WAN edge (for example, often a Cisco 7200 Series router) is connected to the core via interfaces that do not support the setting for jumbo frames. It may also usually be a valid assumption that frames received from the remote locations across the WAN are not full 1500-byte sizes. For example, typical deployments use IPsec + GRE over the WAN, so the frames are usually already reduced of size to be carried over the tunnels. Thus, the fact that the MTU size cannot be increased on the WAN edge devices for interfaces connecting to the core may not be a problem. However, this is not the case for traffic coming from the core of the campus and directed toward the WAN edge, so configuring jumbo frame supports on these interfaces may still be required (as shown in Figure 78). The different MTU size setting on the two side of the link prevents the creation of the OSPF adjacency. To work around this issue, the following specific command needs to be issued on the interfaces of the WAN devices:

```
interface FastEthernet1/0
    description Link to campus core
    ip address 10.122.5.101 255.255.255.254
    ip ospf dead-interval minimal hello-multiplier 4
    ip ospf mtu-ignore
```

Doing so instructs the OSPF process running on the WAN edge device to not consider the MTU value as a criterion for the establishment of OSPF adjacency with the core routers.

- Use the **mpls mtu** interface command

  Configuring a value of 1508 on all the MPLS-enabled interfaces allows for transmission of full 1500-byte sized IP packets, because the two additional labels are not considered when comparing the size of the frame to the MTU of the physical interface. The following configuration is enabled on all the MPLS-enabled interfaces of the network (both on the PE and P devices):

```
interface TenGigabitEthernet1/1
 tag-switching mtu 1508
 tag-switching ip
```

Note Note that the “mpls” part of the command is automatically changed to “tag-switching” on Catalyst 6500 platforms.

The main advantage of this approach as compared to the one discussed in the previous bullet is that the MPLS MTU setting does not affect the establishment of routing adjacencies when deploying OSPF. Therefore, this is the recommended approach.

Tagging or not-Tagging Global Table Traffic

The use of network virtualization in the context of this guide is positioned as an evolutionary overlay design that results in much of the traffic remaining in the global table; users or devices are selectively removed from the global routing table to be part of the defined VPNs to solve specific problems (guest/partner access, NAC remediation, and so on).

When MPLS is enabled on the physical link connecting each PE device (in the distribution layer of each campus distribution block) to the high speed core, all the traffic flowing in the network starts to be tagged. The global table traffic uses a single MPLS tag, whereas all the packets related to VPN traffic are characterized by an internal VPN tag and an outer IGP label.

One possible option is then to modify this default behavior and to start tagging only the VPN traffic, leaving all the communications in global table untagged. There are several advantages in doing this:

- MTU—Traffic in global table does not have any of the MTU issues previously discussed because no tags are added to the original packet.
- Troubleshooting—Because global traffic is IP switched and not label switched, this means that all the typical troubleshooting tools can be used to verify the functionalities of global table traffic. There is no requirement to understand the MPLS-specific tools that are discussed in MPLS-Specific Troubleshooting Tools, page 139.
- QoS—As previously discussed, after traffic is tagged with an MPLS label, there are three bits in the MPLS header (the EXP bits) that can be used for carrying QoS information. This allows supporting up to eight classes of traffic, so in the specific situations where the enterprise has already implemented a QoS strategy based on the use of more than eight classes, not tagging the global traffic helps in not disrupting such strategy. Traffic in global table continues to be classified and marked in the same way it was before MPLS VPN was turned on and no changes in the queuing strategy need to be put in place in the overall network.

From a convergence perspective, there is actually not much difference between the scenarios where global traffic is tagged or not. The main factor contributing to the convergence time (in a box/link failure scenario) is IGP convergence; the LDP component is negligible.

In summary, the main advantage of not tagging global table traffic is that the creation of the virtual network becomes a process that is not disruptive to the functionalities already in place in the enterprise network. This functions well with the initial design principle that virtualization should be used to address specific problems and should not affect the majority of the “normal” enterprise communications.

The question now becomes determining the best solution to implement untagged global table traffic. The recommended option is to influence the way LDP exchange tags between the various network devices (P and PE) in the MPLS network. Each device is locally assigning a label to each prefix contained in the global routing table. This functionality is triggered as soon as an interface is configured for label switching and cannot be stopped. What is possible to do instead is to control which labels can be exchanged between neighbor devices through the LDP protocol.

Because VPN routes are exchanged via MP-BGP, they are learned on each PE with a next hop pointing to the loopback of the remote PE device (part of the distribution block where that specific VPN subnet resides). This means that all VPN traffic flowing across the MPLS core is label-switched along an LSP logically terminating on the loopback interfaces that the remote PEs use for MP-iBGP peering. The idea is then to control the distribution of MPLS labels so that only labels associated to the loopback IP addresses are actually exchanged between network devices, which causes only the VPN traffic to be label-switched while all the global traffic is handled as “regular” IP traffic.

To control the distribution of labels between devices, it is possible to apply ACLs to LDP. Using loopback interfaces for establishing LDP sessions allows the clear identification of prefixes for which tags should be advertised. In addition, the recommendation is to assign IP addresses to the loopback interfaces taken from a specific separate range than the addresses assigned to the various campus subnets; in this example, all the loopbacks are addressed from the range 192.168.100.0/24.

The following two configuration steps achieve this purpose:

- Configure the ACL to identify the subnet assigned to the loopbacks:

```
ip access-list standard loopbacks_only
permit 192.168.100.0 0.0.0.255
deny   any
```

- Apply the ACL to LDP

```
no mpls ldp advertise-labels
mpls ldp advertise-labels for loopbacks_only
```

Note It is first required to configure the switch to not advertise any label (**no mpls ldp advertise-labels**) and then to apply the specific ACL.

The **mpls ldp advertise-labels** command can potentially also allow specifying the neighbor to which the allowed labels should be sent. This is shown as follows:

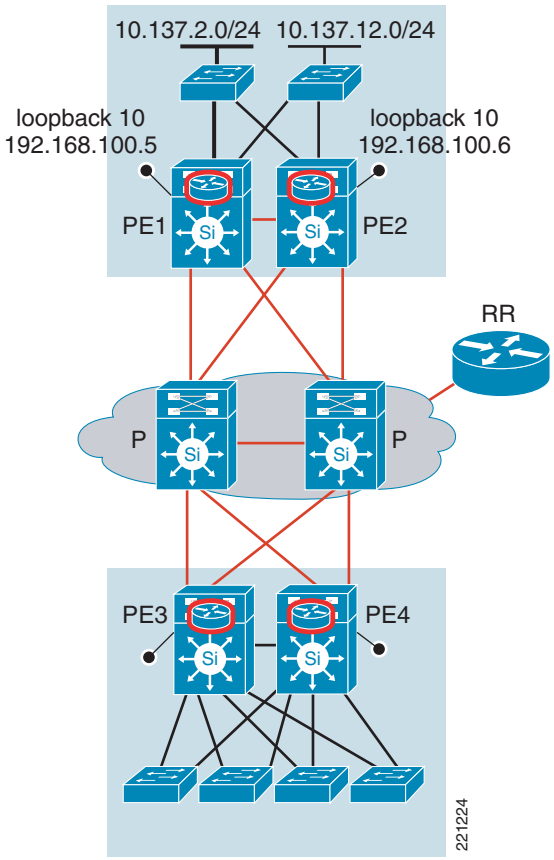
```
cr20-6500-1(config)#mpls ldp advertise-labels for loopbacks_only ?
  to  Access-list specifying controls on LDP peers
  <cr>
```

However, in the example scenario, there is no need to add this additional tuning, because there is the requirement to advertise the label for the loopbacks to all the LDP neighbors.

In addition, the configuration shown above is generic enough to be easily applied on all the MPLS-enabled devices (P and PE), which is an operational advantage.

A practical example verifies this filtering functionality, based on the network topology shown in Figure 79. 10.137.2.0/24 and 10.137.12.0/24 are two subnets defined in a remote distribution block (the former belonging to the global table, the latter part of a VPN).

Figure 79 Label Example



No labels are used to send traffic toward the remote global subnet (note that a summary route 10.137.0.0/16 is actually known on PE3/PE4), as you can see from the "Untagged" keyword in the "Outgoing" tag column.

```
PE3#sh mpls forwarding-table 10.137.2.0
Local  Outgoing    Prefix              Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
67     Untagged    10.137.0.0/16     0          Te1/2       10.122.5.26
       Untagged    10.137.0.0/16     0          Te1/1       10.122.5.30
```

Different considerations are valid for VPN traffic; as shown in the following, the remote VPN subnet is learned via MP-BGP:

```
PE3#show ip route vrf v1 10.137.12.0
Routing entry for 10.137.12.0/24
  Known via "bgp 64000", distance 200, metric 0, type internal
  Last update from 192.168.100.6 02:15:34 ago
  Routing Descriptor Blocks:
  * 192.168.100.5 (Default-IP-Routing-Table), from 192.168.100.1, 02:20:34 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
    192.168.100.6 (Default-IP-Routing-Table), from 192.168.100.2, 02:15:34 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

This means that all the VPN traffic destined for the remote subnet 10.137.12.0 is sent out on the LSP built toward the loopbacks of PE1 and PE2. As expected, the traffic sent to reach these BGP next hops is actually tagged:

```
PE3#sh mpls forwarding-table 192.168.100.5
Local  Outgoing    Prefix              Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
64     32          192.168.100.5/32  0          Te1/2       10.122.5.26
       61          192.168.100.5/32  0          Te1/1       10.122.5.30
PE3#sh mpls forwarding-table 192.168.100.6
Local  Outgoing    Prefix              Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
49     34          192.168.100.6/32 0          Te1/2       10.122.5.26
       62          192.168.100.6/32 0          Te1/1       10.122.5.30
```

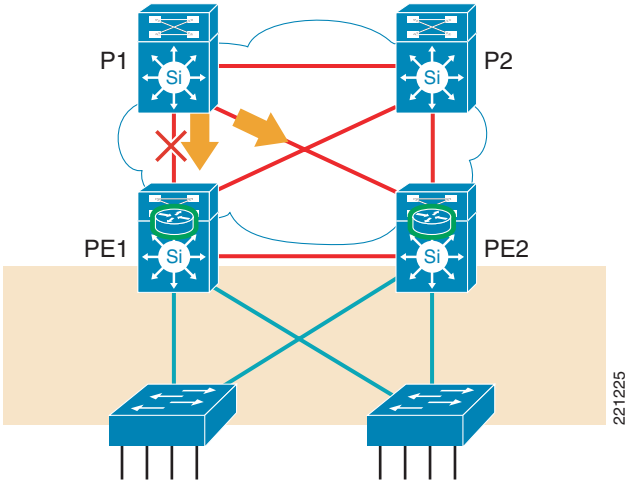
Convergence Analysis for VPN and Global Traffic

When deploying campus networks, the need to support applications such as VoIP that are very sensitive to delay and packet drops demands stringent requirements for convergence under several failure scenarios. It is therefore important to verify that when MPLS VPN is turned on in such an environment, the convergence that is achieved in global table (where voice services are recommended to be deployed) is not affected. Also, depending on the requirements of the applications deployed in the context of each logical partition created in the network, it is also important to verify what kind of convergence can be achieved inside each defined VPN.

Various device-based or link-based failures can occur in a campus network. When deploying MPLS VPN, this maps to a failure of PE, P, or CE devices and their interconnections. The impact of these events is largely dependent on the specific network topology. When building campus networks, the recommendation is usually to implement a high degree of redundancy. For example, this implies creating a full mesh of connections between each specific campus distribution block and the high speed core, and also a full mesh of connections between the core devices themselves when possible. These recommendations hold true also for MPLS VPN deployments, because the main factor affecting traffic convergence for both global and VPN traffic flows in a campus deployment is the IGP convergence. The influence of this factor can be minimized by building equal cost paths between different areas of the network.

Figure 80 shows a scenario where equal cost multi-path (ECMP) is the main factor dictating the convergence under failure scenarios. Note that the recovery mechanism provided by ECMP is valid for both VPN and global table traffic.

Figure 80 Use of ECMP for Convergence



In the equal cost path core configuration, the switch has two routes and two associated hardware CEF forwarding adjacency entries. Before a link failure, traffic is being forwarded using both of these forwarding entries. On the removal of one of the two entries, the switch begins forwarding all traffic using the remaining CEF entry. The time taken to restore all traffic flows in the network depends only on the time taken to detect the physical link failure and to then update the software and associated hardware forwarding entries. The key advantage of the recommended equal cost path design is that the recovery behavior of the network is both fast and deterministic. The one potential disadvantage in the use of equal cost paths is that it limits the ability to engineer specific traffic flows along specific links. Overriding this limitation is the ability of the design to provide greater overall network availability by providing for the least complex configuration and the fastest consistent convergence times.

Note More details on designing campus network for minimizing traffic outage under various failure scenarios (and corresponding convergence results) can be found in the campus design guides at the following URL:
http://www.cisco.com/en/US/netsol/ns656/networking_solutions_design_guidances_list.html#anchor2

The only failure scenario that deserves a more detailed discussion in the context of this guide is the PE device failure, because it is the only case where convergence for VPN traffic is affected by BGP functionalities.

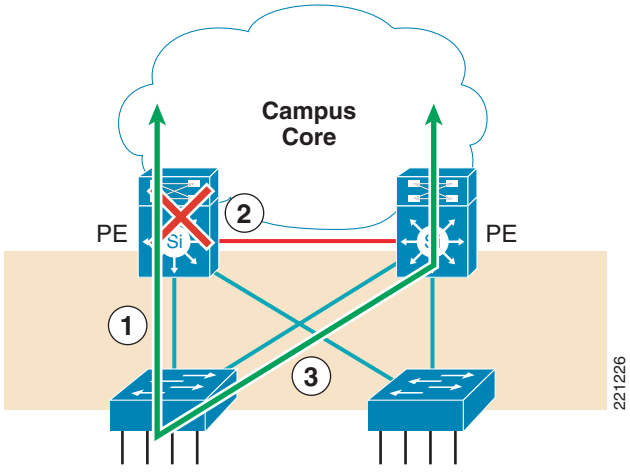
PE Failure

The distribution layer devices play the PE role when deploying MPLS VPN in a multilayer campus. The failure of the PE devices affects convergence for both upstream (that is, leaving the campus distribution block) and downstream (that is, destined to a subnet defined inside the specific distribution block) flows. The upstream and downstream flows are considered separately to better clarify the network elements involved in the recovery.

Upstream Convergence

As shown in Figure 81, the failure of the PE device affects the upstream traffic that needs to start flowing through the second PE device.

Figure 81 Upstream Convergence with PE Failure



The sequence is as follows:

1. Traffic is flowing through the left PE switch (representing the HSRP active device).

2. The PE fails.
3. Traffic is redirected to the second PE switch, which has become the new HSRP active device.

The main factor influencing the convergence in this scenario is the HSRP recovery time. Following the design recommendation of implementing sub-second HSRP timers (as discussed in Virtualization of Network Services at the Distribution Layer, page 98) allows achieving sub-second convergence (actual convergence time is around 700–800 msec).

Note the additional considerations for this specific scenario:

- Upstream recovery time is the same for global and VPN flows because the HSRP mechanism is common to both types of traffic.
- The recovery time is independent from the specific connectivity type between the distribution block and the high speed core. This means that it is the same in fully meshed, partially meshed, or ring topologies.
- The recovery time is independent from the specific IGP (EIGRP or OSPF) running in global table and from MP-BGP governing the exchange of VPN routes.
- The failure of the PE switch functioning as the HSRP standby device has no impact on upstream traffic convergence.

Downstream Convergence

This is a more complex scenario when compared to upstream convergence because the factors affecting the recovery are different for traffic in global table or in the context of a specific VPN. First consider traffic flowing in global table; there may be various scenarios, depending on the type of connectivity between the distribution block and the high speed core, as shown in Figure 82.

Figure 82 Downstream Convergence with PE Failure

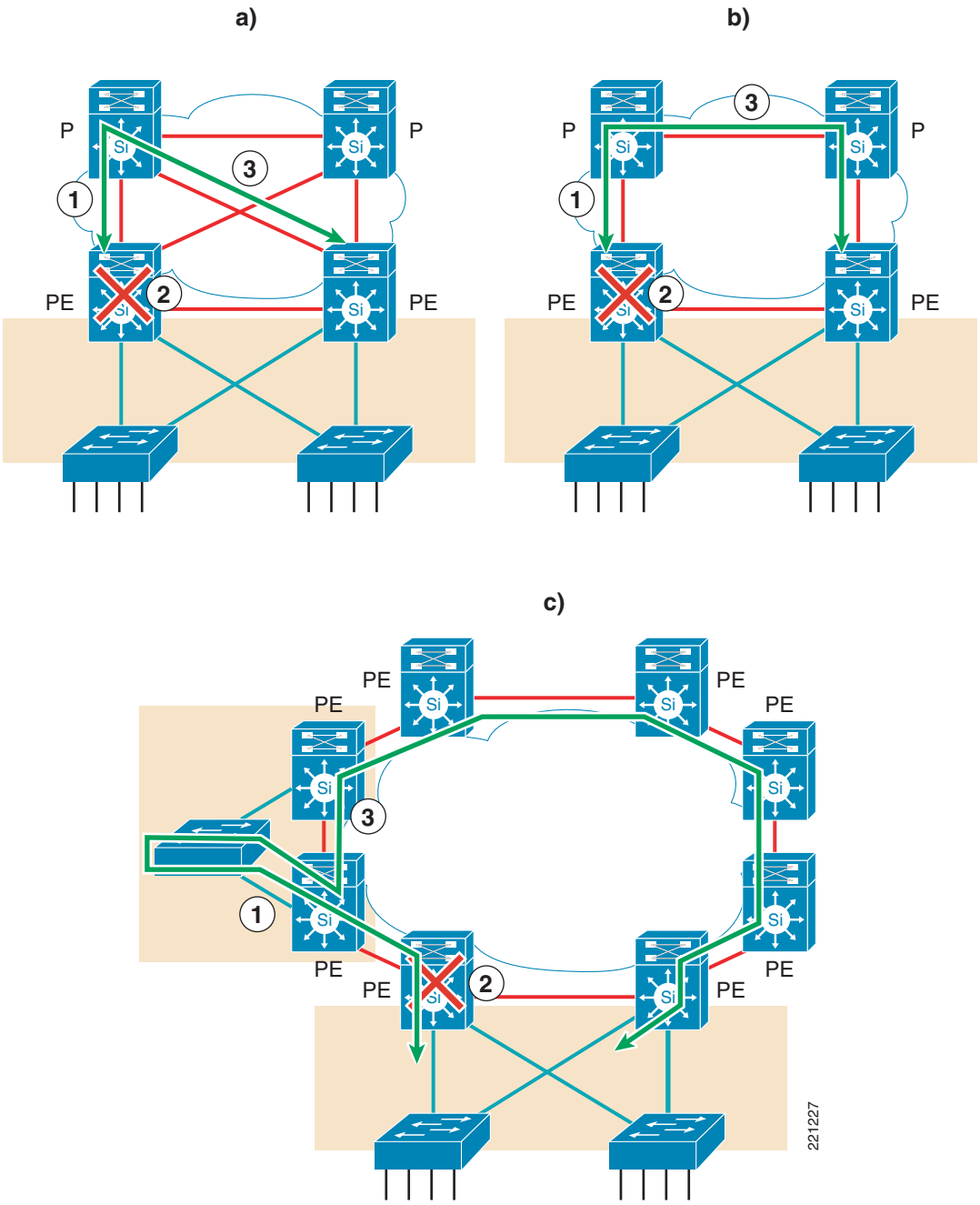


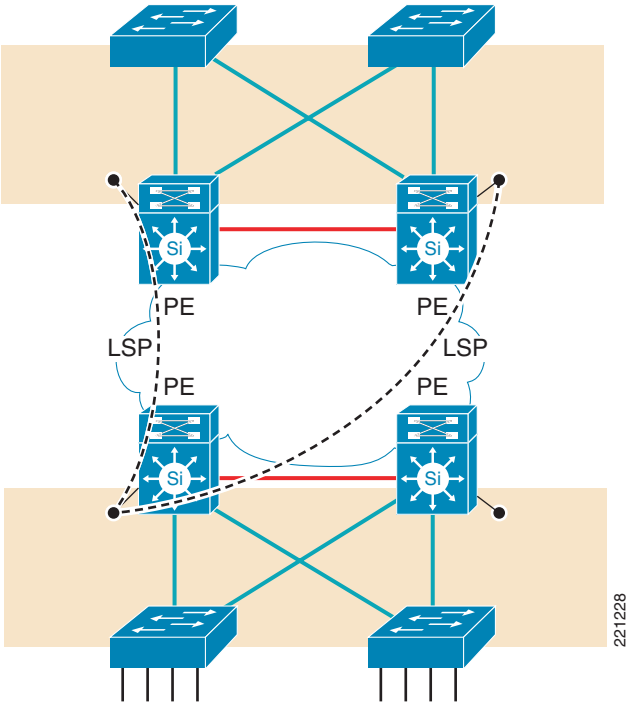
Figure 82 shows the following scenarios:

- In the fully meshed scenario (a), the core devices have a redundant equal cost path connecting to the campus distribution layer. This means that restoration of downstream traffic after a distribution switch failure is achieved using Layer 3 equal cost recovery in the core switches. Therefore, traffic recovery depends on the capability of the core device to reprogram the hardware CEF to start using the other adjacency already in place (when the failure of the link connecting to the PE is detected). This process is very fast and usually allows recovery in less than 200 msec

- In the partial mesh scenario (b), the mechanism that allows recovering global table flows is the IGP rerouting. Traffic that was originally sent from the core to the failing PE needs now to be rerouted across the link connecting the core devices and down through the second PE belonging to the distribution block. Using sub-second timers for the IGP (EIGRP or OSPF) allows keeping the convergence in the same order of 200–300 msec.
- In the ring scenario (c), the mechanism allowing the recovery is once again IGP rerouting, so the convergence is similar to that discussed in the previous point.

Various considerations need to be made for the recovery of VPN traffic. In this case, the main factor in the reestablishment of the traffic flow is MP-BGP convergence. Figure 83 shows this specific scenario.

Figure 83 LSP Establishment

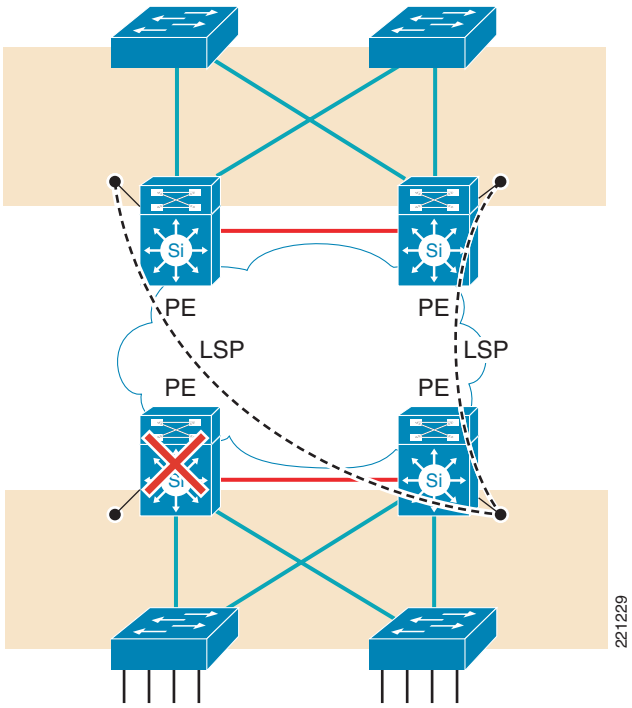


Before the PE fails, any pair of remote PEs (belonging to a different campus distribution block) establishes an LSP connecting the loopback interfaces defined on each device (the VPN traffic is label switched between PEs using the information contained in the external MPLS label). This is because, as discussed in a previous section, all the VPN routes are exchanged via MP-BGP, and thus are imported into each VRF routing table with the next hop specifying the loopback of the remote PE (directly connected to the remote VPN subnet), as follows:

```
cr20-6500-1#sh ip route vrf v1 10.138.12.0
Routing entry for 10.138.12.0/24
  Known via "bgp 64000", distance 200, metric 0, type internal
  Last update from 192.168.100.9 10:07:08 ago
  Routing Descriptor Blocks:
  * 192.168.100.9 (Default-IP-Routing-Table), from 192.168.100.1, 10:07:08 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

When the PE fails, traffic originated in the above distribution block needs to be rerouted via a different pair of LSPs, connecting to the second PE device of the remote distribution block, as shown in Figure 84.

Figure 84 PE Failure



The configuration is as follows:

```
cr20-6500-1#sh ip route vrf v1 10.138.12.0
Routing entry for 10.138.12.0/24
  Known via "bgp 64000", distance 200, metric 0, type internal
  Last update from 192.168.100.10 00:00:06 ago
  Routing Descriptor Blocks:
  * 192.168.100.10 (Default-IP-Routing-Table), from 192.168.100.1, 00:00:06 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Note In a fully-meshed design, each PE receives an equal cost path route for the VPN subnets from each remote PE. In that scenario, each PE sends traffic along LSPs connecting to both remote PEs, which also causes half of the traffic to be blackholed in case of remote PE failure. Apart from this, all the following considerations are applicable independently from the specific topology (fully meshed, partially meshed, or ring).

The sequence of events leading to the switching of LSPs is as follows:

1. The PE fails and the IGP in global table notifies all the devices that the loopback of the failed PE is no longer reachable. The corresponding entry is removed from the global routing table.
2. Until the BGP scanner runs on the various PEs, the entries pointing to the failed PE are maintained in the BGP table. This means all the traffic directed to it is blackholed.
3. When the BGP scanner runs (by default, this happens every 60 seconds), a check for the next-hop address is performed (in global table) and, because the loopback of the failed PE is gone, the BGP table is updated. This also triggers the update of the hardware CEF so that traffic can resume through the redundant remote PE.

Because of the sequence described above, a failure of a PE device may cause a worse case outage of 60 seconds for the VPN traffic.

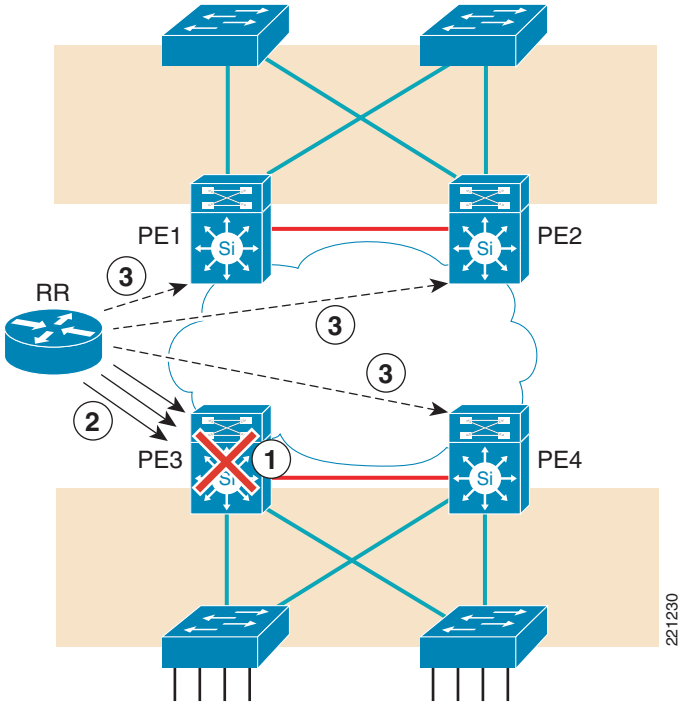
Worse-case scenarios may arise where a default route is injected in global table. This is not typical in service provider environments, but may become an important factor in MPLS VPN campus deployments where the use of default route is quite common.

The root of the problem is that when doing the next-hop checking (at step 3 above), BGP considers that default route as a viable path to the failed PE, even if the more specific loopback is removed from the routing table. As a consequence, if for some reason (for example, because of redistribution) the metric associated to the default route is "better" than the metric associated to the loopback of the redundant remote PE, the old entry pointing to the failed PE is maintained as valid in the BGP table after running the BGP scanner. This implies that all the traffic is blackholed at this point, which remains valid until the holdtime for the failed PE expires and the route reflector notifies the PEs to withdraw the routes learned from the failed PE. This can lead to a worst case outage of up to 180 seconds (the default BGP holdtime value). Therefore, the recommendation is to ensure that the IGP metric associated to the default route is always higher than the metrics associated to the PE loopback interfaces used for establishing iBGP peering connections.

Even after fixing the issues associated with the existence of the default route, there is still the worst-case convergence scenario of 60 seconds previously described. The following are several ways of reducing this outage:

- The first solution is to use a feature called next-hop tracking. This workaround is not currently available for Catalyst 6500 platforms (it will be included in the next IOS release), and it is thus positioned for campus WAN edge deployments where Cisco 7200 platforms can be deployed as PE devices. This feature introduces an event-driven notification system to monitor the status of routes that are installed in the routing database and to report next-hop changes that affect internal BGP (iBGP) prefixes directly to the BGP process. This improves the overall BGP convergence time by allowing BGP to respond rapidly to next-hop changes for routes installed in the routing database, instead of waiting for the periodic BGP scanner to run. This specific case is discussed in more detail in Extending Path Isolation over the WAN, page 141.
- A second solution is to tune down the timer so that the BGP scanner runs more frequently. The minimum configurable value is 5 seconds, which allows achieving a best-case convergence of 5 seconds. However, tuning the BGP scanner timer should be done carefully, because depending on the number of routes present in the BGP table, this may significantly affect the performance of the CPU and is therefore not the recommended solution.
- The recommended way of improving BGP convergence in the context of this guide suggests tuning down the BGP timers (keepalive and holdtime), so that the failing of the PE device can be detected quickly and the other PE devices can be notified. Figure 85 shows this concept, assuming the use of route reflector devices (which is the recommended approach as discussed in MP-iBGP Deployment Considerations, page 108).

Figure 85 Reducing Convergence Time by Tuning BGP Timers



The sequence of the events is as follows:

1. PE3 fails and stops sending keepalives to the RR.
2. The RR keeps sending keepalives to the failed PE3.
3. After the holdtime expires (usually three times the keepalive timer), the RR declares PE3 dead and informs the other PEs. The PEs, after receiving the RR notification, remove any prefix previously learned from the failed PE from the BGP table.

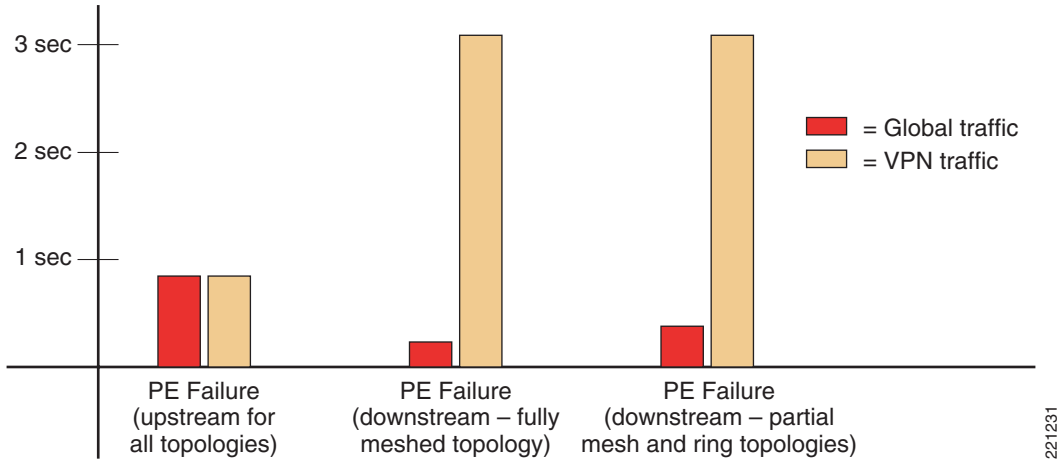
The factor that dictates the convergence time in this scenario is therefore the BGP holdtime. Tuning the timers aggressively (for example, using 1 sec for keepalives and 3 seconds for holdtime) allows reducing the outage to a worst-case scenario of 3 seconds. However, such an aggressive tuning should be carefully considered because it may cause a "false negative" when three consecutive keepalive messages are lost; for example, because of congestion and a PE is declared failed even if that is not the case. This should not be an issue on campus networks characterized by a high speed backbone core, which is why tuning these BGP timers is considered the recommended design practice here.

From a configuration standpoint, the tuning of the timers can be achieved as follows:

```
router bgp 64000
 neighbor 192.168.100.3 remote-as 64000
 neighbor 192.168.100.3 update-source Loopback10
 neighbor 192.168.100.3 timers 1 3
```

Figure 86 summarizes the convergence values for upstream and downstream flows in relation with the specific network topology in place. Note that these values are independent from the specific IGP used (EIGRP or OSPF).

Figure 86 Convergence Values



Summary of Design Recommendations

Based on what has been discussed in the previous sections of this guide, the following summarizes the best practice design recommendations when deploying MPLS VPN in a campus environment:

- Each campus distribution block should be connected in a fully-meshed fashion to the high speed core devices to improve the convergence (for both global table and VPN traffic) under various failure scenarios. When possible, the core devices should also be fully meshed between them.
- PE devices should be positioned at the first L3 hop of the campus network, represented by the distribution layer devices in a multilayer campus design.
- LDP and MP-iBGP peering should be implemented using loopback interfaces. When deploying OSPF as IGP, these loopback interfaces should be defined in the specific OSPF area deployed in each distribution block.
- The loopback interfaces used for iBGP and LDP peering should be addressed from a separate and well-identifiable IP subnet.
- LDP targeted hellos should be used to improve the convergence in link/box recovery scenarios.
- Summarization of subnets from each distribution block should be implemented for global table prefixes but not for VPN routes.
- Given the high level of symmetry found in the recommended campus network designs, the following configuration steps should be implemented to achieve load balancing of VPN traffic:
  - The PE devices belonging to the same distribution block should use unique Route Distinguisher values.
  - MP-BGP configuration should make use of the “maximum-paths ibgp 2” command
- BGP timers (keepalive and holdtime) should be tuned down to <1 sec, 3 sec> to reduce the outage for VPN traffic in case of PE device failures.
- The **mpls mtu** command should be configured on all the MPLS-enabled interfaces to avoid fragmentation issues.

MPLS-Specific Troubleshooting Tools

When turning on MPLS VPN, a new set of troubleshooting tools needs to be used to verify the proper functionalities of the network. Unless it is decided not to tag traffic in global table (as discussed in Tagging or not-Tagging Global Table Traffic, page 127), the troubleshooting techniques need to focus on two aspects: MPLS traffic (for global table) and VPN traffic.

MPLS Troubleshooting

- Verify that the proper interfaces are MPLS-enabled and that the right Label Distribution Protocol (LDP in this example) is configured:

```
PE1#sh mpls interfaces
Interface              IP            Tunnel   Operational
TenGigabitEthernet1/1  Yes (ldp)     No       Yes
TenGigabitEthernet1/2  Yes (ldp)     No       Yes
TenGigabitEthernet1/3  Yes (ldp)     No       Yes
```

Note The “Tunnel” field refers to the capacity of Traffic Engineering for each specific interface.

- Verify that LDP neighbors are discovered out of all the MPLS-enabled interfaces and that the proper LDP Identifier (loopback address) is configured for this device:

```
PE1#sh mpls ldp discovery
 Local LDP Identifier:
    192.168.100.5:0
    Discovery Sources:
    Interfaces:
        TenGigabitEthernet1/1 (ldp): xmit/recv
            LDP Id: 192.168.100.19:0
        TenGigabitEthernet1/2 (ldp): xmit/recv
            LDP Id: 192.168.100.18:0
        TenGigabitEthernet1/3 (ldp): xmit/recv
            LDP Id: 192.168.100.6:0
```

  If an LDP neighbor is not discovered out of one of the interfaces that are MPLS-enabled, first verify the IP connectivity between the loopback interfaces of the neighbor devices. Remember that without IP connectivity, the TCP session between loopbacks fails and the LDP session cannot be established.

- Verify the assignment of labels to each prefix contained in routing table:

```
PE1#sh ip route
Codes: 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, E - EGP
       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
Gateway of last resort is 10.137.0.3 to network 0.0.0.0
D       10.122.5.100/31
          [90/3328] via 10.137.0.3, 00:09:40, TenGigabitEthernet1/3
          [90/3328] via 10.122.5.30, 00:09:40, TenGigabitEthernet1/1
C       10.137.2.0/24 is directly connected, Vlan2
<SNIP>

cr20-6500-1#sh mpls ldp bindings 10.122.5.100 31
  tib entry: 10.122.5.100/31, rev 558
        local binding:  tag: 44
        remote binding: tsr: 192.168.100.6:0, tag: 94
        remote binding: tsr: 192.168.100.19:0, tag: 21
```

```
cr20-6500-1#sh mpls ldp bindings 10.137.2.0 24
  tib entry: 10.137.2.0/24, rev 568
        local binding:  tag: imp-null
        remote binding: tsr: 192.168.100.6:0, tag: imp-null
```

As shown in the output above, for each entry in global routing table learned via IGP from a neighbor, the device generates a local tag (which is then advertised to the neighbor devices), and also receives tags from the LDP neighbors. For each directly connected subnet (as it would be for a locally generated summary), the device allocates an implicit NULL tag (and receives one from the peer switch also directly connected to the subnet). This is done to instruct the neighbor devices to perform PHP for all traffic destined to that subnet.

- Verify that the LSP connecting global subnets defined in different distribution blocks is functional. This can be done by using the **traceroute** command. For example, assuming that a remote global subnet is 10.138.2.0/24,he output of the command is as follows:

```
cr20-6500-1#traceroute 10.138.2.1
Type escape sequence to abort.
Tracing the route to 10.138.2.1
  1 10.122.5.30 [MPLS: Label 50 Exp 0] 0 msec 0 msec 0 msec
  2 10.122.5.13 [MPLS: Label 63 Exp 0] 0 msec 0 msec 0 msec
  3 10.122.5.43 4 msec *  0 msec
```

As seen above, the communication along the LSP is successful and the MPLS tag used at each hop is also shown.

Note The MPLS traceroute functionality works differently than traceroute on a normal IP network. For more details on this, see the following URL: http://www.cisco.com/warp/public/105/mpls_traceroute.pdf

MPLS VPN Troubleshooting

- Verify that VRF is properly defined (route distinguisher, route-target communities, and so on), and the right interfaces are associated to it.

```
cr20-6500-1#sh ip vrf detail v1
VRF v1; default RD 64001:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Vlan12                  Vlan13                  Vlan14
    Vlan15                  Vlan16                  Vlan17
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:64000:1
  Import VPN route-target communities
    RT:64000:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 16)
```

A different command can be used to ensure that the specific interfaces are properly mapped to the VRF and actually removed from global table:

```
cr20-6500-1#sh ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
Vlan12                 10.137.12.3     v1                               up
Vlan13                 10.137.13.3     v1                               up
Vlan14                 10.137.14.3     v1                               up
```

```
Vlan15                 10.137.15.3     v1                                  up
Vlan16                 10.137.16.3     v1                                  up
Vlan17                 10.137.17.3     v1                                  up
```

- Verify the control plane: VPN routes should be properly received from the remote PE devices:

```
cr20-6500-1#sh ip bgp vpnv4 vrf v1
BGP table version is 2149, local router ID is 192.168.100.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 64001:1 (default for vrf v1)
* i10.136.0.20/31   192.168.100.3            0    100      0 ?
*>i                 192.168.100.3            0    100      0 ?
* i10.136.0.100/31  192.168.100.4            0    100      0 ?
*>i                 192.168.100.4            0    100      0 ?
<SNIP>
```

- Verify the data plane: use the **traceroute** command to verify connectivity in the context of each defined VPN:

```
cr20-6500-1#traceroute vrf v1 10.138.12.1
Type escape sequence to abort.
Tracing the route to 10.138.12.1
  1 10.122.5.30 [MPLS: Labels 61/24 Exp 0] 0 msec 0 msec 0 msec
  2 10.122.5.13 [MPLS: Labels 18/24 Exp 0] 0 msec 4 msec 0 msec
  3 10.138.12.3 0 msec *  0 msec
```

Note how both the MPLS labels are now shown at each hop: the external label is modified (61 becomes 18), while the internal VPN label remains untouched because it is used by the receiving PE to properly switch the packet in the right VPN.

Extending Path Isolation over the WAN

Overview

Several options must be considered when extending path isolation beyond the campus and across the WAN. Selecting the proper approach is based on the number of branches that constitute the WAN and the number of required virtualized networks. This section describes three approaches that each represent a balance in design between complexity and robustness. Smaller networks that require only a small number of virtualized networks do not require the same level of complexity as a larger enterprise with many virtualized networks.

Design Options—Three Deployment Models

The following three deployment models can be used to provide path isolation over the WAN:

- Multi-VRFs on the WAN edge and branch edge mapped to SP-provided L3 VPN service (profile 1)
- Multi-VRFs mapped to DVMPN tunnels (profile 2)
- RFC 2547 over an L2 VPN service (profile 3)

The following subsections describe the advantages and disadvantages of each of these deployment models.

Initial Conditions

Before describing these three possible deployments, the following assumptions concerning initial conditions are made:

- The first assumption is that there is an existing network, and that path isolation is required to maintain separation between the current infrastructure and a new network or networks that overlay the current infrastructure.

  This covers the majority of situations. If no initial network exists, the methods presented here can still be deployed, but some initial thought should be given to determine the purpose of the global underlying network.

  Expanding on this first assumption, the global network continues to support its current role. Virtualized networks should not impose new limitations on the current network. Because the virtual networks are overlaid on top of existing infrastructure, many aspects of the virtual segments have already been locked in place. For example, if the existing WAN is a hub-and-spoke topology, any virtualized network inherits that same property. The robustness of the current routing environment has an impact on the stability of any overlaid network. The same statement applies to the security policy. As a general rule, all virtual networks are restricted by the same limitations that apply to the global network.

- The second assumption is that the existing network has a homogeneous routing domain.

  If the current network is already logically partitioned into multiple autonomous routing systems, the complexity of overlaying a single autonomous domain requires a specialized solution. This is an extension of the fact that virtualized networks inherit many attributes from the underlying infrastructure. Of particular concern are networks that employ a BGP core. Overlaying a new BGP cloud over an existing one, or proposing a method to integrate RFC 2547 into an existing domain, is not covered in this release of network virtualization.

The focus of this design guide is limited. For example, multicast running inside of a VRF may be a completely feasible scenario that can be deployed using common PIM design principles; however, no testing has yet been done by Cisco to confirm that there are no interactions between VRFs and PIM. As a result, no guidance is given to cover this type of deployment. Current guidance is focused on IP connectivity, and includes routing protocols as well as some supplemental services such as Cisco IOS firewall.

Enterprise MPLS Terminology

Three WAN path isolation deployment models based on VRFs are considered. The first two are based on multi-VRF CEs and the third is based on RFC 2547 capability that is extended to edge of the enterprise edge network. Before exploring these three methods, some clarification on the terminology is helpful.

Historically, MPLS has been deployed by service providers. Common MPLS terms are used to define the role of a device as it is used within the SP network. Routers known as P or LSR nodes have no customer-facing interfaces. This differs from PE or LER routers, which are at the edge of the service provider network, and the CE router, which is at the edge of the customer network. In this guide, these terms are useful only as a method to determine the function of a device within the MPLS framework. As mentioned in previous sections, P nodes swap labels, PE nodes are the gateway between label switching and IPv4 routing domains, and CE devices are purely IP routing nodes, although they may be partitioned onto logical IP domains, or VRFs. Because this guide is focused solely on the enterprise network, the standard terms are somewhat ambiguous. The terminology becomes especially confusing when interfacing with a service provider that is offering an MPLS service. In such a situation, a device may take on a dual role depending on the perspective. A router can be a CE from the service provider

perspective, but serve in a PE role within the enterprise environment. In addition, consider a device that is a CE to both the provider and the enterprise. The same term has two meanings, depending on perspective.

In an effort to alleviate some of the confusion, this chapter avoids the use of the term “provider” when speaking of an enterprise-owned and enterprise-operated device. Instead, a device is referred to as an enterprise label switch, enterprise label edge, or a multi-VRF router. Devices referred to as PEs or Ps are only those that belong to the service provider network, and are out of scope of this guide.

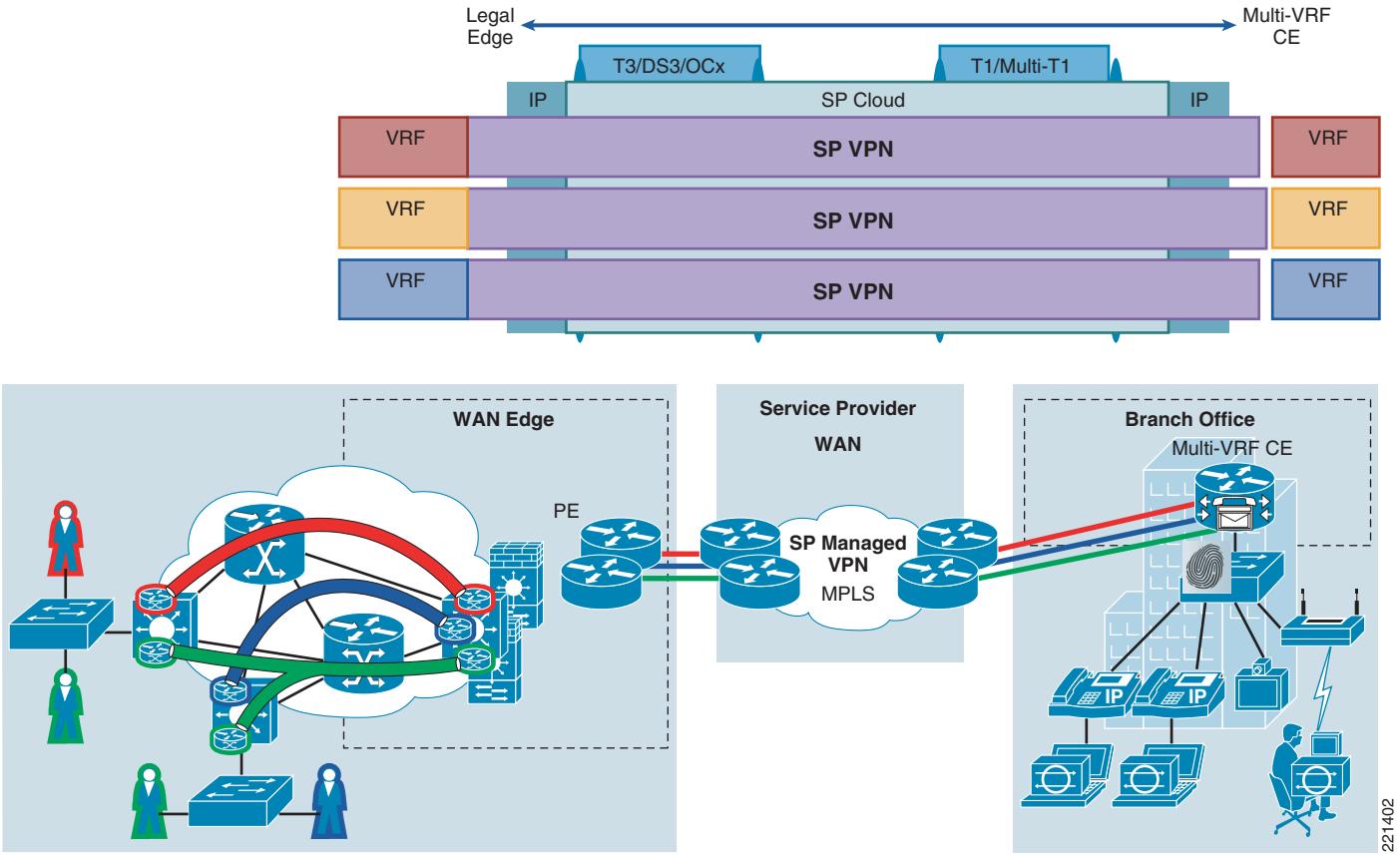
The main difference lies in the ownership of the labels. SP devices interact only with SP-created labels. Similarly, enterprise devices work only with enterprise-created labels. Within MPLS, there is a feature known as Carrier supporting Carrier (CsC). In this setup, one MPLS provider may switch labeled traffic transparently over another MPLS provider. Some carriers are beginning to offer this type of service to enterprise customers. In the deployment models considered here, the outer header of a packet is always an IP field when transitioning between SP and enterprise MPLS clouds. It is helpful to keep this in mind when considering the WAN path isolation techniques described in the following sections.

Mapping Enterprise VRFs to Service Provider VPN (Profile 1)

The first method to extend path isolation across the WAN is useful when only a small number of VRFs need to be transported, such as from one to five VRFs plus the existing global table. The enterprise WAN aggregation routers server as the enterprise label edge device. The branch employs multi-VRF routers. No enterprise labels are transported across the SP network. Instead, the enterprise label edge router maps VRFs directly into a VPN service that is offered by the service provider. This service is most likely an L3-based VPN. At the branch side, each VPN service is mapped to unique VRFs that are contained solely within the branch router. These VRFs are then mapped directly to VLANs. If multiple routers exist at the branch location, path isolation is maintained by mapping VRFs onto VLANs that are then transported over an Ethernet trunk to downstream devices.

Figure 87 shows an example of this deployment model.

Figure 87 Mapped SP-Provided MPLS VPNs



This method is attractive because it is easy to set up and straightforward to troubleshoot because the enterprise labels are confined to the campus. This simplicity also enhances the solutions availability. No additional control plane is required. However, the approach is restricted by the cost of the MPLS VPN service and by the Layer 2 handoff between the enterprise and service provider domains. In addition, the lack of a dedicated control plane leaves the solution somewhat static. Adding or removing VRFs from the network requires a concentrated effort and coordination with the service provider.

There are the following two starting points to consider:

- The existing global WAN is running over an MPLS-provided L3 VPN.

  This is becoming more common. This service is often based on MPLS. In this situation, the standard MPLS terminology can become unhelpful, as mentioned above.

- The global table is not already in an SP-provided VPN.

  In this case, it is possible to subscribe to a service for *only* the VRFs without moving the global traffic. However, another port on the branch router is required for the isolated WAN links. This is not an ideal situation because the VRFs are transported via an L3 VPN, while the global table may be transported over an L2 VPN such as Frame Relay. This situation is possible, however, because each VRF is mapped directly to some VPN service.

Connecting the Enterprise to the Service Provider

The most common handoff to an SP-provided VPN is Ethernet, although any L2 transport can be used. Some providers can support trunking on this connection. This is the lowest cost option for both the SP and the enterprise. If the existing link cannot support L2 VLANs, such dot1Q trunks, additional physical connections are required for each VRF. This can become prohibitively expensive very quickly. A simple configuration for this setup is as follows:

```
!
interface FastEthernet0
 description Existing VPN service connection
 ip address 192.168.200.29 255.255.255.252
!
interface FastEthernet0.101
 encapsulation dot1Q 201
 ip vrf forwarding v1
 ip address 192.168.101.2 255.255.255.252
!
interface FastEthernet0.102
 encapsulation dot1Q 202
 ip vrf forwarding v2
 ip address 192.168.101.6 255.255.255.252
!
interface FastEthernet0.103
 encapsulation dot1Q 203
 ip vrf forwarding v3
 ip address 192.168.101.10 255.255.255.252
!
```

Note that the original interface has not been disturbed. This satisfies the goal of not modifying the base network to support virtualization. In this situation, the global traffic is sent without an 802.1Q tag. The main interface remains in the global table. Some customers may be more comfortable moving the subnet from the main interface onto a sub-interface. There is nothing wrong with this. The same reasoning likely applies to the LAN interface as well. However, note that the scale limitation in terms of the number of VRFs supported is actually imposed by the number of VLANs the branch router can handle, because each L3 VRF maps to an L2 VLAN at the branch router. This depends on the platform and the version of Cisco IOS.

```
router_1811(vlan)#vlan 29
Vlan can not be added. Maximum number of 28 vlan(s) in the database.
```

If the link between the enterprise router and the SP router is a point-to-point trunk, it is beneficial to disable spanning tree on the associated VLAN.

QoS on the WAN Interface

When an enterprise customer selects SP-managed L3VPN service to interconnect sites, it is not uncommon to find a hierarchical QoS configuration on WAN interfaces because the actual upstream bandwidth is typically below the interface speed. This is done to create an artificial back pressure to allow packet reorder as well as traffic shaping. Because shaping is not allowed simultaneously at the physical interface and sub-interfaces, it is normally best to shape traffic only at the main interface. The existing QoS configuration is shared among all VRFs. If the global table has been moved onto a sub-interface, mixing the service policy between the main interface and sub-interfaces is not required. This allows unique service policies for each sub-interface. There are several limitations: class-based weighted fair queueing (CBWFQ) is not supported on sub-interfaces, and the class “class-default” is not VRF aware. There is only a single default class. The following example shows a typical three-class model for the global traffic, and a specific policy applied to one VRF.

```
!

interface FastEthernet0

!
interface FastEthernet0.100
 description Existing VPN service connection
 encapsulation dot1Q 200
 ip address 192.168.200.29 255.255.255.252
 service-policy output T1
!
interface FastEthernet0.101
 encapsulation dot1Q 201
 ip vrf forwarding v1
 ip address 192.168.101.2 255.255.255.252
 service-policy output KIDS_ROOM
!
interface FastEthernet0.102
 encapsulation dot1Q 202
 ip vrf forwarding v2
 ip address 192.168.101.6 255.255.255.252
 service-policy output T1
```

This is a relatively complex configuration that shows a rather trivial example. However, the objective is to prevent the traffic in one VRF from consuming all the bandwidth. This becomes an important part of DoS containment. No guarantees can be made at present. A virus in one VRF can have a detrimental impact on the traffic in adjacent VRFs. The example above is limited. There is another, more complex approach to this problem. Object tracking can be used across VRF boundaries. This means that an IPSLA probe can be set up in one VRF and then a corrective action applied in an adjacent VRF. Consider the following configuration:

```
!
track 50 rtr 50
!
!
ip route vrf V4 0.0.0.0 0.0.0.0 192.168.101.13 track 50
ip route vrf V2 10.173.255.255 255.255.255.255 Null0 track 50
ip route vrf V4 0.0.0.0 0.0.0.0 Null0 250
!
!
no ip http server
no ip http secure-server
!
ip sla 50
 udp-jitter 10.173.112.1 32768 source-ip 192.168.101.6 num-packets 6 interval 50
 threshold 15
 vrf V2
!
ip sla reaction-configuration 50 react jitterAvg threshold-value
15 10 threshold-type consecutive 2 action-type triggerOnly
!
ip sla schedule 50 life forever start-time now
!
!
!
```

In this example, an IPSLA probe is set up to measure the jitter to a particular destination in VRF V2. If the jitter exceeds a given threshold, the routing table in VRF V4 is adjusted. This method can protect traffic in VRF V2 from the traffic in VRF V4. In this case, two examples are shown; the first places the default router in V4 onto an alternate path. Another option would be to place a default to null in VRF V4. In both cases, the objective is to move lower priority VRFs off the shared physical link. Unlike the first example, the probe is able to react to traffic in both directions. However, the corrective action is

applied only outbound via the local routing table. In advanced cases, it is possible to inject a dummy route into the routing protocol and then track and react to this route at another point in the network. This method is compressive because it can track the performance in one VRF in both directions, and react to the event at multiple points in the network over multiple VRFs, all with basic Cisco IOS.

Routing within a VRF

The physical topology of individual VRFs is very similar to the underlying global VRF within the confines of the enterprise network. However, this does not mean that each VRF is strictly limited to the underlying structure., because each VRF is mapped into a SP L3 VPN. In most cases, all VRFs attach to the same campus. However, it is possible that the campus in one VRF may not be the same as another. In addition, a campus may not have presence in all VRFs across the entire population of branches. It is certainly possible and probable that only a subset of the total branches appear in a given VRF. In this guide, only the condition where all VRFs share the same campus and the same set of branches is considered. Hybrid situations must be custom-designed.

Because the relationship in topologies between VRFs is only loosely dependant, each VRF should run a routing protocol. Normal design principles apply. Routes should be summarized as much as possible. The use of stub areas and default routing should be used whenever practical. In most situations, the pre-existing global table has implemented BGP peering with the service provider.

There are several ways to set up routing over an MPLS-VPN service, such as using unique AS numbers at all enterprise sites, using AS-Override with SoO checking to prevent loops, or using simple default routing. The approach used in the global table should be replicated in each of the VRFs. If BGP peering is used on the PE-CE link, this should be adequate to provide routing for smaller one- and two-router branches. The following is a sample configuration for the global table plus two additional VRFs:

```
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 192.168.200.2 remote-as 65535
 !
 address-family ipv4
 redistribute connected metric 100
 redistribute static
 neighbor 192.168.200.2 activate
 no auto-summary
 no synchronization
 network 10.173.1.0 mask 255.255.255.0
 network 192.168.200.0 mask 255.255.255.252
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute connected
 neighbor 192.168.201.237 remote-as 65535
 neighbor 192.168.201.237 activate
 no synchronization
 network 10.173.25.0 mask 255.255.255.0
 exit-address-family
 !
 address-family ipv4 vrf V2
 redistribute connected
 neighbor 192.168.201.241 remote-as 65535
 neighbor 192.168.201.241 activate
 no synchronization
 network 10.173.24.0 mask 255.255.255.0
 exit-address-family
```

The campus configuration is more complex. The WAN router may be peering with multiple service providers to allow a more complete service area, competitive pricing, and some level of redundancy for critical branch locations. Again, the simplest approach is to use the global configuration as a template.

If the enterprise is not currently subscribed to an MPLS service, a more detailed look at the routing implications is recommended, such as the *L3 MPLS VPN Enterprise Consumer Guide Version 2* at the following URL:
http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a008077b19b.pdf

Scale Considerations

The following two limiting factors determine the scalability of profile one:

- Cost is likely the deciding factor in determining how many VRFs are feasible in most situations. This cost is associated to SP charges for the additional VPNs. Cisco has tested this profile up to five VRFs. This number is somewhat arbitrary and is not a hard limit.
- The number of VLAN a branch router can support. Each VRF requires two VLANs; one for the WAN connection, and another for the LAN. The branch routers have a software-imposed VLAN limit, which can be determined with the **show vtp status** command as follows:

```
Branch_2>sh vtp status
VTP Version                     : 2
Configuration Revision          : 40
Maximum VLANs supported locally : 12
```

  Subtract 4 from this number to account for the special use VLANs. Because each VRF requires two VLANs, and one VLAN is needed as the default global VLAN, this box is software-limited to three VRFs. The number of supported VLANs is a consideration in all three profiles; however this is the only profile that requires two VLANs per VRFs.

The routing protocols and associated route tables also represent a load on the CPU. These are discussed in General Scalability Considerations, page 158.

Multiple VRFs Over a Single VPN (Profile Two)

It may not always be possible or practical to procure additional VPN networks from the service provider. The costs can become prohibitive, or the media in place may not readily allow parallel paths, which is the case with a dedicated leased line T1 circuit. If it is not possible to provide L2 path isolation, the enterprise is required to run L3 tunnels over the single L2 transport. Each tunnel can then be placed into a unique VRF.

This is similar to the previous approach, and has the following advantages:

- No additional services are required from the SP. This keeps the solution cost-effective on a per-VRF basis by allowing enterprise customers to resize the VPN network without any deployment cost impact.
- The L3 tunnels may terminate on a dedicated box. The enterprise label edge is not required to be the WAN aggregation box as was the case in the first profile.
- The tunnels may be encrypted.

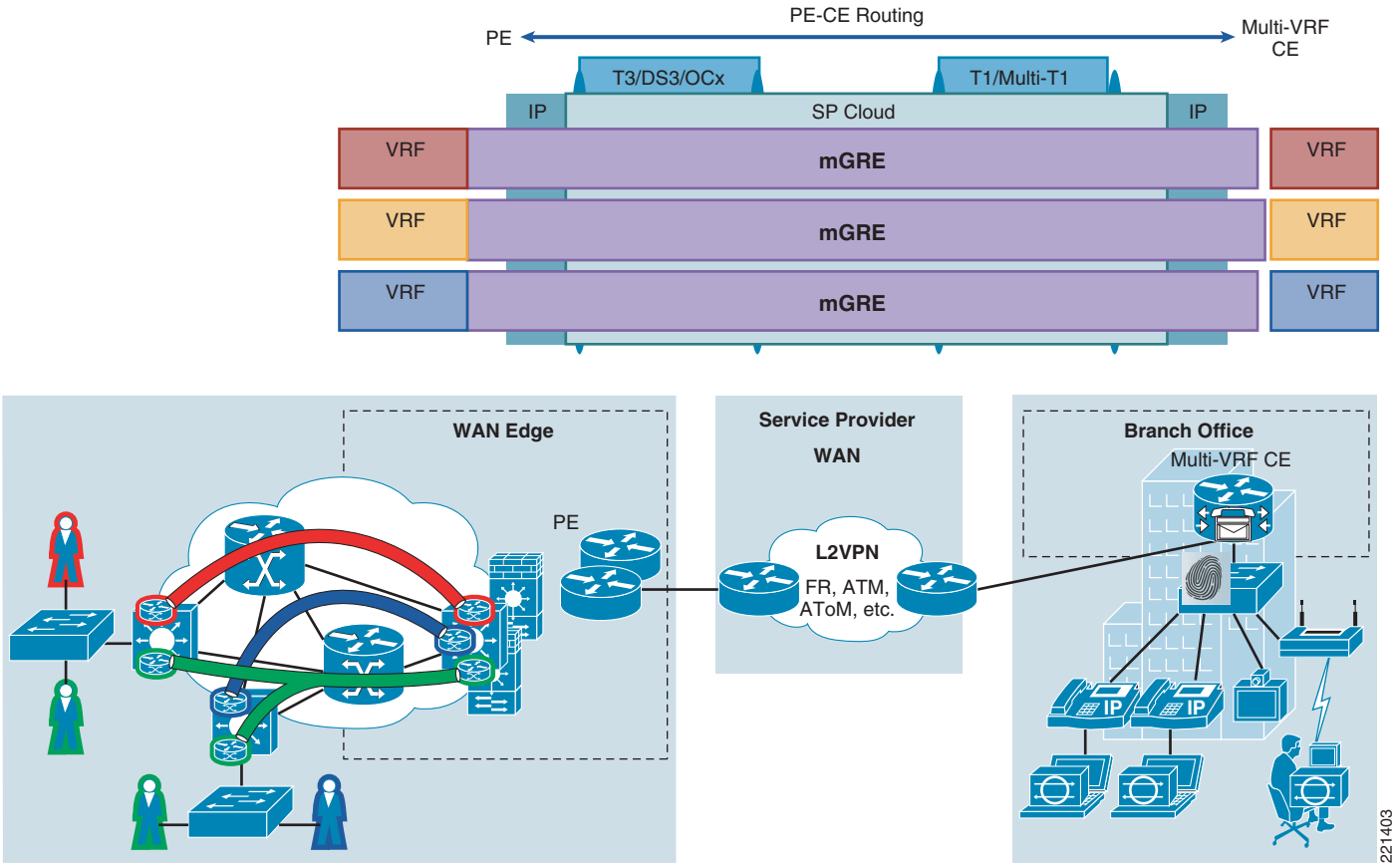
This deployment model also has the following restrictions:

- The tunnel configurations can grow to be quite large and complex.
- Additional overhead is required to provide tunnel end-to-end connections.

- Additional headers must be placed onto each packet. Keep in mind that because the enterprise label edge device resides at the campus, no tags are placed on packets inside the tunnel.

The recommended tunneling method for this profile is DMVPN (see Figure 88). This offers an encrypted service and minimizes the configuration on the WAN aggregation router.

Figure 88 Mapped DMVPN over a Common IP Cloud



Isolation versus Privacy

MPLS-based VPNs do not natively encrypt their payloads. Privacy is maintained only by wedging a tag into the packet header. This allows a unique and logically isolated routing environment. MPLS can be considered a protocol that provides isolation without guaranteeing privacy. A network analyzer has little difficulty viewing the data from any VRF if the operator can obtain the packet. This can happen either accidentally through a provision mistake, or intentionally through a malicious attack. The first line of defense is always physical security, which ensures that only trusted support people are allowed physical access to the equipment and circuits. The next level is to harden the devices to prevent a remote attack. The final line of defense is to encrypt the packets so that the data is safe even if physical security is breached. In some situations, it is a legal requirement that any packet transported off the premises must be encrypted. Profile two is the only design that includes payload encryption.

MPLS with DMVPN

The basic setup is to create a unique DMVPN tunnel for each required VRF. The tunnels transport packets from the enterprise label edge to a multi-VRF branch router. The WAN aggregation device is somewhere in the path, but not necessarily at the enterprise label edge. The outside address space of the tunnel is in the existing global routing domain. The inside address space of the tunnel is inside the isolated VRF. Although it is possible to place the outside address of the tunnels within a VRF, this rarely required.

The following is the basic configuration of the tunnel as seen on the branch router:

```
!
interface Tunnel10
 ip vrf forwarding v10
 ip address 10.173.160.3 255.255.255.0
 ip mtu 1400
 ip nhrp authentication secret
 ip nhrp map 10.173.160.1 10.173.255.3
 ip nhrp network-id 10
 ip nhrp nhs 10.173.160.1
 ip ospf hello-interval 30
 load-interval 30
 tunnel source GigabitEthernet0/0
 tunnel destination 192.168.200.5
 tunnel key 10
!
```

Note that a network ID is used to provide NHRP with a unique identifier, and a tunnel key is used to provide a unique identifier to the IPsec tunnel. These are required when multiple tunnels are using the same pair of outside addresses. However, the VPN SPA as seen in a Catalyst 6500 does not support the tunnel key option. Therefore, it may be necessary to set up a unique head-end address for each tunnel. The branch router can still source all tunnels from the same WAN interface address, and can also use unique loopback addresses. The only requirement is that the address is reachable in the global table. In addition to restrictions on the tunnel key, the older VPNSM found in the Catalyst 6500 does not allow VRF-based DVMPN tunnels (see CSCek64643). At the time of this writing, careful consideration should be given when deploying this profile in an environment where the Catalyst 6500 serves as the DVMPN hub.

IPsec tunnels used to transport encrypted packets are also established in the global table. Typically, tunnel profiles are used in DMVPN deployments. This is one example when crypto maps make more sense. With crypto maps, it is possible from multiple DMVPN tunnels to share the same IPsec tunnel. Tunnel profiles attempt a unique IPsec tunnel per mGRE interface. This can dramatically limit the scalability of DVMPN per VRF deployments. Defined crypto maps applied to the physical interface allow more control to handle the many parallel DVMPN tunnels. A typical configuration is as follows:

```
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key BIGSECRET address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 10
!
!
crypto ipsec transform-set NV esp-3des esp-sha-hmac
!
crypto map SP1_BUNDLE 10 ipsec-isakmp
 set peer 192.168.200.17
 set transform-set NV
 match address HEAD_END1
crypto map SP1_BUNDLE 20 ipsec-isakmp
 set peer 192.168.200.13
```

```
 set transform-set NV
 match address HEAD_END2
!
crypto map SP2_BUNDLE 10 ipsec-isakmp
 set peer 192.168.200.5
 set transform-set NV
 match address HEAD_END1
crypto map SPA_BUNDLE 20 ipsec-isakmp
 set peer 192.168.200.9
 set transform-set NV
 match address HEAD_END2
```

This example shows a large branch that is connected via two service providers, shown as SP1_BUNDLE and SP2_BUNDLE. Within each provider, the branch can reach two head-end routers, shown as HEAD_END1 and HEAD_END2. These maps are then applied to the respective interfaces; either the SP1 interface or the SP2 interface. Each bundle is transporting ten DMVPN tunnels, each forwarding a unique VRF. Each of the ten DMVPN tunnels is mapped to the primary and backup head-end router.

This configuration is difficult to deploy based solely on tunnel profiles. The design provides the branch flexibility to route individual VRF traffic to a favored head-end over a favored SP connection. Head-end redundancy can be provided in the following two ways:

- Single DMVPN cloud, multi head-end—Define two NHS for each tunnel; one per head end.
- Multi-DMPVN cloud, multi head-end—Create a unique tunnel mapped to each head end.

In small single-attached branches using tunnel profiles, the single DMVPN cloud is typically used. When multi-attached branches are used, or when crypto maps are used, the multi-DMVPN cloud, multi head-end approach tends to be more common. The first approach offers a simplified configuration. The second allows better convergence. With the multi-tunnel approach, a standard routing protocol can determine availability over the inside network. With a single tunnel, NHRP handles failures via hellos traversing the outside address space. In a virtualized environment, the multi-tunnel approach is preferred.

Note Note in the example that a wildcard mask is used on the pre-shared key. This is not considered a secure practice, but does allow easy deployment in a test environment. In an actual production environment, certifications should be used, or the key mask restricted to known addresses. Further discussion on these topics can be found in the respective design guides.

Routing Over VRF-Mapped DVMPN Tunnels

Of the three profiles, profile 2 is the only one that supports an end-to-end IGP to operate inside VRF and over the WAN at the same time. There are some obvious advantages to this. First, the enterprise can send routing hellos through the path isolation structure. The timing of these hellos can be determined by the enterprise. In many cases, the routing protocol running inside the tunnel detects and responds to a service interruption before the outside address space has converged. When comparing this solution to the first solution, no BGP peering is required to support the VRF traffic within the WAN environment. BGP may still be used in the campus to transport VRF information.

The two common routing protocols that are considered are EIGRP and OSPF.

EIGRP Running in a VRF

When deploying EIGRP, there are some general rules concerning EIGRP over a DVMPN cloud. The routing protocol considers the DVMPN subnet to be a multi-access media. It assumes peer-to-peer connectivity that is not always in place. Although DVMPN supports spoke-spoke tunnels, it is often advantageous to run DMVPN in a hub- and-spoke-only model.

If DMVPN is going to be run in a full spoke-spoke mode, **no ip next-hop-self eigrp <*as*>** should be configured to allow EIGRP to pass the next hop information without alteration. The reason is that even though EIGRP believes Spoke_A and Spoke_B are on the same subnet and are therefore adjacent, the spokes do not maintain a persistent connection to one another.

If DVMPN is used in a hub-and-spoke-only mode, **no ip split-horizon eigrp <*as*>** should be configured to allow the hub to rebroadcast the routing information from one spoke to the next. It is also best practice to configure EIGRP as a stub such that each branch receives only a default route from the hub.

Although each VRF maintains a unique EIGRP topology, they are run under the same process. This means that a single instance of EIGRP is configured, and then address-families are set up to handle the VRFs. Each address-family is assigned its own AS number that is advertised to neighbors within that VRF. The router configuration is similar to the following example:

```
!
router eigrp 173
 no auto-summary
 !
 address-family ipv4 vrf V1
 network 10.0.0.0
 no auto-summary
 autonomous-system 173
 exit-address-family
 !
 address-family ipv4 vrf V2
 network 10.0.0.0
 no auto-summary
 autonomous-system 173
 exit-address-family
 !
 address-family ipv4 vrf V3
 network 10.0.0.0
 no auto-summary
 autonomous-system 173
 exit-address-family
 !
 address-family ipv4 vrf V4
 network 10.0.0.0
 no auto-summary
 autonomous-system 173
 exit-address-family
 !
```

Troubleshooting EIGRP within a VRF is handled the same way it would be in the global domain. The only difference is the addition of the VRF in the show command. For example, to view the EIGRP topology of V3, the command “show ip eigrp vrf V3” should be used.

OSPF Running in a VRF

Unlike EIGRP, OSPF runs a unique process for each VRF. Two key commands are required to configure OSPF. First, under the routing process, the command **capability vrf-lite** disables label edge checks intended to prevent mutual redistribution loops. Because the branch router is not processing labels, these checks are not needed and they prevent OSPF from functioning properly in a multi-VRF branch router.

To allow proper OSPF function over the DMVPN interface, the hub interface should be configured as an **ospf point-multipoint** interface. The branch routers then need to adjust their OSPF hello timers to match the hub interface. By default, this is 30 seconds. Details on running OSPF on a DMVPN interface can be found in the *DVMPN Design Guide*.

```
!
interface Tunnel11
 ip vrf forwarding v1
 ip address 10.173.171.6 255.255.255.0
ip ospf hello-interval 30
!
interface Tunnel12
 ip vrf forwarding v2
ip ospf hello-interval 30
!
interface Tunnel13
 ip vrf forwarding v3
ip ospf hello-interval 30
!
!
!
router ospf 171 vrf v1
 log-adjacency-changes
 capability vrf-lite
 network 10.0.0.0 0.255.255.255 area 1
!
router ospf 172 vrf v2
 log-adjacency-changes
 capability vrf-lite
 network 10.0.0.0 0.255.255.255 area 1
!
router ospf 173 vrf v3
 log-adjacency-changes
 capability vrf-lite
 network 10.0.0.0 0.255.255.255 area 1
!
```

Scale Considerations

The size of the network and the number of VRFs supported is a function of the hub router. Virtualization does not add much direct overhead because the labels are removed before reaching the WAN infrastructure. However, VRFs allow a multiplication factor that must be considered. Consider a 1000-node WAN that is used to support four VRFs. This can result in an effective load on the hub equivalent to a 4000-node network in the control plane, and potentially a 4000-node network if the VRFs are used to support new end users. The total number of end users determines data plane loading because this is strictly a function of PPS rates on each device.

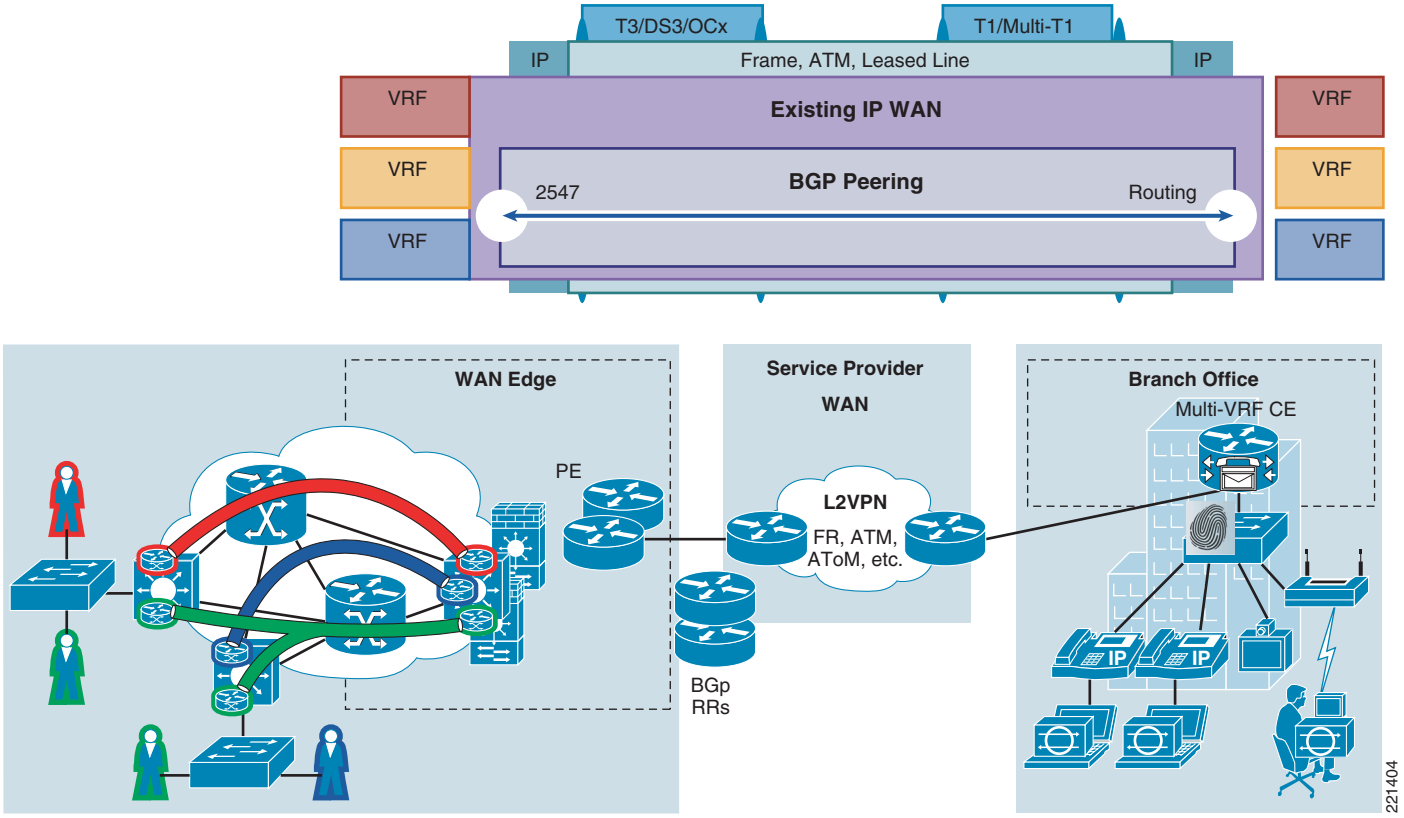
The control plane load can be reduced by overloading the crypto tunnels. This is done by using a single crypto tunnel to carry all the GRE tunnel traffic rather than creating multiple parallel tunnels. The previous example shows this setup. The crypto tunnels are in the global table.

Cisco has tested this profile up to ten VRFs. The number is arbitrary. Beyond ten VRFs, the configurations become large, and troubleshooting multiple parallel DMVMN tunnels becomes difficult. If more than ten VRFs are required, consider extending the enterprise label edge to the branch router (profile 3).

Extending the Enterprise Label Edge to the Branch (Profile 3)

The final WAN path isolation design extends label switching to the branch routers. This provides the most flexibility in terms of adding and removing VRFs. It also scales higher than the previous two approaches, and results in a smaller configuration because parallel paths over the WAN are not explicitly defined. However, this approach also requires overlaying BGP on top of the existing global WAN, and distributing labels over the WAN. This third profile is based on RFC 2547 running over an existing L2 transport. (See Figure 89.)

Figure 89 RFC 2547 Over an L2 Cloud



The goal of leaving the pre-existing global table undisturbed adds some unique challenges because MPLS does not transport data without tags. The requirements create a hybrid between IP routing for the global traffic, and label switching for the traffic using VRF. Both methods run over the same WAN links. Knowing which packets to tag with labels and which to route directly requires proper planning.

Note The information in this guide represents a minimum base knowledge required to implement this profile. For more information on MPLS VPN concepts, see the *Layer 3 MPLS VPN Enterprise Consumer Guide* at the following URL: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a008077b19b.pdf

Setting up BGP over the WAN

The first step is to build an iBGP structure over the internal WAN. New loopback addresses are required. These addresses should be easily classified with an access list. This can be best accomplished by assigning all loopback subnets from the same supernet. These subnets are distributed in the global table to allow BGP peering. They cannot be summarized within the routing table between label edge nodes. These loopback subnets are also used to determine which packets are switched via labels and which are routed as part of the existing global table.

Ideally, the loopback interfaces and associated subnets are created specifically for the task of WAN path isolation and are not shared with other tasks, such as DLSw+ peering and so on. This facilitates the label distribution access list used later in the configuration. The level of detail in planning at this stage determines how successful WAN isolation will be. The network administrator should consider how many branches may become involved in the future. Sufficient space should be allocated in the address space, and the structure should be easily understood.

Route Reflector Placement

Scalable multi-protocol iBGP requires route reflectors (RRs), which are used to allow end-to-end iBGP connectivity without having to build a fully-meshed, direct peering structure. Their placement should be such that they can reach all branches that participate in the global table. Typically, they are deployed close to the WAN edge. They may be inline or out-of-line. Inline route reflectors are in the data path, and are often the WAN aggregation routers themselves. This is usually seen only in small-scale networks. In modest- to larger-sized networks, the route reflectors are placed as one-arm bandits just off the data path. The only function of these devices is to disseminate route information to the iBGP speakers. The route reflectors also need a loopback interface. The loopback subnets are added to the global routing table. However, as mentioned previously, it is important not to summarize these routes. The route reflectors peer with each branch and also with the campus route reflectors. The WAN and campus route reflectors can be collapsed; however, in this situation, churn in the WAN topology is not compartmentalized from the campus. Load on the route reflectors as the result of a WAN convergence event can have a detrimental impact on campus VRF stability.

Integration of Campus and WAN Route Reflectors

Because the WAN route reflectors and campus route reflectors are best deployed as dedicated servers, some discussion on interconnectivity is appropriate. Because the MPLS cloud should be end-to-end between the data center and the branch LANs, a single AS should be used on both campus and WAN RRs. The peering between the RRs must be fully meshed. The routes between the peers should be stable and free of redistribution between IGPs, such as multiple IGPs. In some environments, it may be possible to directly connect the WAN and campus RRs via a DWDM pipe or single cable, depending on the physical distance between the boxes. Any instability in the fully-meshed peering between the RRs is felt on all VRFs over the end-to-end enterprise environment.

Label Distribution

A key part of this solution is extending the label edge to the branch. This is based on RFC2547. However, a goal of this design is to push labels only onto packets that belong to a virtual segment. Packets from the global table are not label switched. This permits existing outbound QoS service policies to continue to function. To configure VRF-only switching, an access list is used to isolate the loopback addresses of the label edges from the other prefixes in the global table. If all branch loopbacks can be summarized as mentioned previously, the same access list can be used throughout the label edge routers. The basic configuration of the branch router is as follows:

```
!
mpls label protocol ldp
no mpls ldp advertise-labels
mpls ldp advertise-labels for pe_loops
!
!
interface Loopback0
 ip address 192.168.100.57 255.255.255.255
!
!
router bgp 64000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.100.58 remote-as 64000
 neighbor 192.168.100.58 update-source Loopback0
 neighbor 192.168.100.59 remote-as 64000
 neighbor 192.168.100.59 update-source Loopback0
 !
 address-family vpnv4
 neighbor 192.168.100.58 activate
 neighbor 192.168.100.58 send-community extended
 neighbor 192.168.100.59 activate
 neighbor 192.168.100.59 send-community extended
 bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute connected
 no synchronization
 exit-address-family
 .
 .
 .
 address-family ipv4 vrf Vn
 redistribute connected
 no synchronization
 exit-address-family
!
ip access-list standard pe_loops
 permit 192.168.100.0 0.0.0.255
!
```

Although this configuration template is discussed as part of the campus design, some items are repeated here for completeness.

Two control plane functions are independent and complementary: LDP is responsible for distributing label information to adjacent nodes, while BGP is responsible for IPv4 prefix information. Both protocols are required. Two labels are pushed onto each packet. The first represents the destination subnet, while the second represents the next IP hop as described by BGP. A label path must be present between edges of the BGP cloud. The route reflectors do not need to be part of this path if they are out-of-line. All other BGP devices are enterprise label edge devices. The SP cloud in this profile is strictly a Layer 2 cloud and does not participate in MPLS. It is possible to learn routes from BGP without a complete label switch path in place. This is a unique consideration with MPLS troubleshooting.

WAN Convergence

In the event of a failure, routing can be used to determine an alternate path. When two paths are present, the possibility of loops is a concern. Loops can occur for a variety of reasons, such as a control plane disconnect between the edges of the routing cloud. Mutual redistribution between routing protocols is a common reason for this. This is likely to occur when backdoor links are present in a native IGP such as

OSPF or EIGRP that is used at the branch locations. Normally, local IGPs at the branch are not necessary. Additional branch routers downstream of the WAN attached routers can simply be added into the iBGP cloud. If a local branch routing protocol is needed to handle many L3 closet switches, a BGP network statement should be used to pick up a summary route rather than redistribution.

Convergence times with iBGP are slow when compared to traditional enterprise protocols. It is possible to decrease timers to try to improve this. Issues to consider include the fact that decreasing timers increases router work load. Most of the delay in route propagation is a result of the scan time. While this timer is configurable within reason, it is a low priority process. Under a large convergence event, the reduced timers may only offer a modest improvement.

```
WAN-RR1#sh proc | in BGP
 231 ME    FC61D8          2624     6636939       0 5992/9000   0 BGP Router
 232 ME    FB5900          4796      290054      16 4944/6000   0 BGP I/O
 233 Lsi   FBF850          2968      623280       4 6640/9000   0 BGP Scanner
 234 Mwe  102CB40             4         157      25 4048/6000   0 BGP Event
 239 Msa   FB52E0             0       15611       0 5696/6000   0 BGP Open
```

Because most convergence events are localized, a reduced scan timer can be quite effective. Because of the low priority, the timer can be reduced without causing a large negative impact on other processes running on the device.

MTU Considerations

Because the branch router pushes two labels onto the packet, the IP MTU on an MPLS-switched interface should not exceed 1492 bytes to reduce the amount of packet fragmentation.

Many devices now implement an MTU path discover mechanism known as path MTU (PMTU). This is done by setting the DF bit in the packet header. If the packet needs to be fragmented, an ICMP Too Big error is sent back by the MPLS-switched interface with a maximum size of 1492. The sending stack adjusts the next packet to this destination accordingly. In the event that ICMP messages are blocked via an access list, or the client is not capable of discovering the MTU, the local LAN interfaces of the client may be set to less than 1492 bytes.

QoS Features

MPLS labels hide the DSCP value of the underlying IP packet. By default, the three MSB bits of the DSCP value are copied into the three EXP bits of the MPLS header. This is known as uniform mode and is the recommend practice for enterprise MPLS. It is possible to set interface service policies based on this mapping. A modified class map for mission-critical may look similar to the following:

```
class-map match-any MISSION-CRITICAL-DATA
 match ip dscp 25
 match mpls experimental topmost 3
```

Although Cisco IOS offers complete flexibility in QoS policy configurations, the recommendation is to apply QoS policy with regards to application and not with regards to VRFs. Generally, the mission-critical data in VRF V1 should share bandwidth with the mission-critical data in all the other VRFs as well as the global traffic. Because the voice marking is not currently supported inside of a VRF, it may be tempting to reallocate EXP 5 to other applications. The recommendation is to leave this marking available for future use.

These configurations assume that MPLS is running over a private L2 WAN such as leased line. In this case, there is a single end-to-end DiffSev model extending end-to-end over the enterprise network. If an SP MPLS VPN service is used, the SP DiffServ domain may be different from the enterprise domain. Any virtual networks added to the enterprise should follow the model deployed in the global table. This

can be uniform mode, short tunnel mode, or tunnel mode. These tunnels methods allow mappings to be used between the SP DiffServ domain and the enterprise domain. For more details about interfacing with an MPLS VPN SP, see the QoS configuration guide.

Scalability Considerations

This profile has the potential to load the enterprise label switch router at the WAN aggregation beyond levels normally seen in the SP environment. This is because this label switch router can be peered to hundreds of enterprise label edge boxes. Cisco tested this profile with 25 VRFs across 460 branches, each VRF with four subnets. The total network composed of 46,000 routes and 11,500 labels. A packet load was applied to the network, and performance numbers were recorded from the LSR, as shown in Table 4.

Table 4 Testing Results

	CPU %	Kpps	Mbps
7200G1	32	150	770
Sup32	2	150	770

These two devices were set up in an active/standby condition. Each device was failed and the load was forced onto the standby box. Cisco noted failover times of less than 20 seconds.

General Scalability Considerations

All three profiles result in an effective network that is multiple times larger than the original one. There are the following two components with loading considerations:

- Control plane load—The result of routing neighbors, route topology stability, management polling, and so on
- Data plane load—Strictly a function of PPS

The addition of VRFs directly impacts control plane loading and indirectly impacts data plane loading. The control plane load can be conservatively modeled by multiplying the size of the WAN by the number of VRFs. A 300-node network with three VRFs to all locations results in 900 peering relationships. If additional users are placed into new VRFs, the data place load is increased by that traffic. If users are simply moved from the global table into a VRF, the data load is not meaningfully changed. In a stable network, the majority of network load is from the data plane. However, enough processor headroom must be available to the control plane to handle convergence events. When adding VRFs, it is important to remember that virtual networks apply a real load that is mostly seen in the control plane.

Multiple Routing Processes

Each OSPF process is assigned a unique PID, and requests service independently from the scheduler, and separately from other OSPF processes. This implies that the total load would be more than simply the number of VRFs multiplied by the number of neighbors. However, the additional process overhead is offset by the fact that the LSA database is contained within a process, simplifying Dijkstra calculations when compared to a single flat LSA table of the same size. The result is that the rule of multiplying the number of VRFs by the number of neighbors is a conservative approach to modeling the total scale.

EIGRP uses address families to handle VRFs. This means that a single EIGRP process handles all VRFs. In extreme situations where all VRFs are under major convergence, this can result in EIGRP generating CPU Hog messages, or the process voluntarily suspending before all VRFs have converged. Normal tools such as stub networks should be used within VRFs to reduce the number of outstanding active queries.

Branch Services

Cisco IOS has many additional features beyond those required for routing packets, including NAT, IOS firewalls, IPS, DHCP Server, and so on. Some of these features are VRF-aware, some are not, and some impose restrictions when operating in a VRF. These branch features apply to all three WAN path isolation profiles.

IOS Firewall Services

There are two types of IOS firewalls. The classic type is based on IP inspections. However, there is a trend toward zone-based firewalls. Both methods are VRF-aware with the following restrictions:

- Inspections cannot be shared across VRFs. Each VRF needs a unique inspection policy, even if that policy has the same set of rules.
- Zones cannot span VRFs. A VRF can contain multiple zones, but a zone cannot appear in more than one VRF.

These restrictions represent a sound security policy and should not cause any problems for most deployments. The disadvantage is that the Cisco IOS firewall configuration needs to be repeated for each VRF, which increases the size of the configuration file.

IOS IPS

This feature is not supported on VRF interfaces as of Cisco IOS Software release 12.4(9)T.

DHCP Server

This feature is supported with some restrictions. Two pools may not be associated to the same VRF. This restricts the number of LANs in VRF to one if the router is going to a server such as the DHCP server. This restriction does not apply if a centralized DCHP server is used.

WAN Path Isolation—Summary

The following three profiles are proposed to handle WAN path isolation, with each profile appropriate for a specific need and size:

- Profile 1 is appropriate when the customer is already subscribed to an SP-managed Layer 3 VPN service such as MPLS and needs only a small number of VRFs. No encryption is provided with this profile. The payload of packets is easily readable if the SP cloud is compromised. Because this profile is based on MPLS, any-to-any connectivity is possible without loading the enterprise WAN aggregation boxes.
- Profile 2 is appropriate when encryption is required for VRF-based traffic. This profile is recommended when less than ten VRFs are required. Beyond that, the solution becomes overly complex. The profile can be deployed over any existing WAN transport including leased line or

MPLS. Platform restrictions at the head end make the Cisco 7200 Series router the best choice for the DVMPN hub device. Branch-to-branch traffic can be accomplished but increases the potential loading on the branch routers.

- Profile 3 provides the most flexibility for additional VRFs. 25 VRFs have been tested. The Cisco ISR does support basic label edge functionality. This profile does not provide encryption and is only appropriate where the enterprise is using an L2 VPN such as Frame Relay or leased line. Customers already subscribed to an SP-managed L3 MPLS VPN cannot use this profile.

None of the profiles are intended for large enterprise environments that have deployed a BGP core. The addition of VRFs into these environments must be handled with a custom design on a case-by-case basis.

Path isolation on the WAN requires specialized support skills. Customers without a well-trained operations staff may wish to invest in additional training to reduce downtime. VRFs add another dimension to what would normally be a simple WAN problem.