- Multipoint GRE
- IPsec

Hybrid Virtual Private Networks

Rather than just using a single MPLS-based VPN technology or a single tunnel-based VPN technology, you can use select VPN technologies in tandem. For example, you might want to extend an MPLS network at one corporate location to MPLS networks at remote corporate locations, while having a requirement that traffic traveling through a service provider's cloud be encrypted.

You could meet the requirements of such a design by having a Layer 3 MPLS VPN set up over a DMVPN. The DMVPN technology carrying the Layer 3 MPLS VPN traffic allows you to efficiently set up direct links between corporate locations, and it also allows you to use IPsec, which can encrypt the traffic flowing through the service provider's cloud.

When it comes to hybrid VPNs, a significant design consideration is *overhead*. Every time you add an encapsulation, you are adding to the total header size of the packet. With more headers, the amount of data you can carry inside a single packet is decreased. As a result, you might have to configure a lower *maximum transmission unit (MTU)* size for frames on an interface.

MPLS VPN

MPLS VPNs extend the capabilities of MPLS, supporting VPNs created across an MPLS network. These VPNs, most commonly found in service provider or large enterprise networks, can be categorized as either Layer 2 MPLS VPNs or Layer 3 MPLS VPNs.

Layer 2 MPLS VPN

With a Layer 2 MPLS VPN, the MPLS network allows *customer edge (CE)* routers at different sites to form routing protocol neighborships with one another as if they were Layer 2 adjacent. Therefore, you can think of a Layer 2 MPLS VPN as a logical Layer 2 switch, as depicted in Figure 2-1.

Key Topic

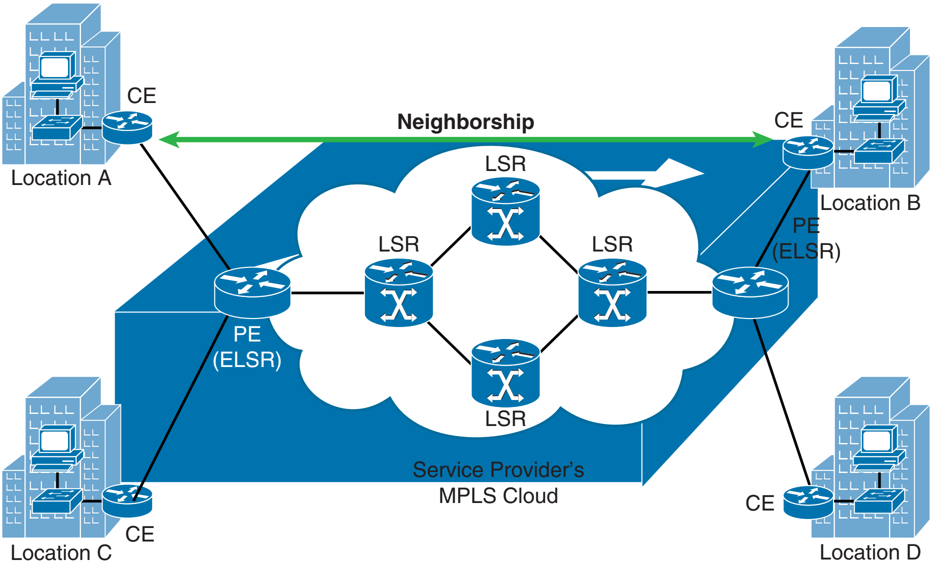

Figure 2-1 *Logical View of a Layer 2 MPLS VPN*

Layer 3 MPLS VPN

With a Layer 3 MPLS VPN, a service provider's *provider edge (PE)* router (also known as an *Edge Label Switch Router [ELSR]*) establishes a peering relationship with a CE router, as seen in Figure 2-2. Routes learned from the CE router are then sent to the remote PE router in the MPLS cloud (typically using *multiprotocol BGP [MP-BGP]*), where they are sent out to the remote CE router.

Key Topic

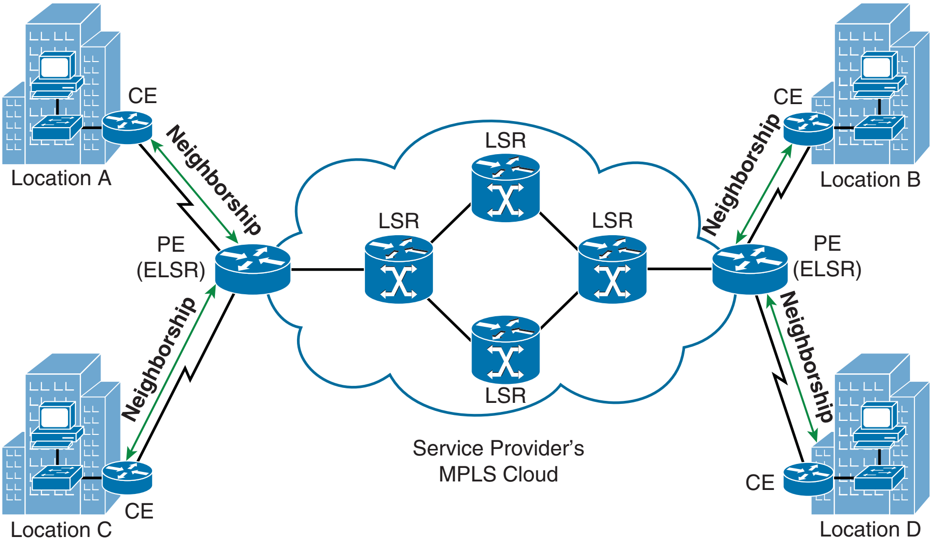

Figure 2-2 *Layer 3 MPLS VPN*

GRE

As its name suggests, a *Generic Routing Encapsulation (GRE)* tunnel can encapsulate nearly every type of data that you could send out of a physical router interface. In fact, GRE can encapsulate any Layer 3 protocol, which makes it very flexible.

GRE by itself does not provide any security for the data it transmits; however, a GRE packet can be sent over an IPsec VPN, causing the GRE packet (and therefore its contents) to be protected. Such a configuration is commonly used, because IPsec can only protect unicast IP packets. This limitation causes issues for routing protocols that use IP multicasts. Fortunately, a GRE tunnel can encapsulate IP multicast packets. The resulting GRE packet is an IP unicast packet, which can then be protected by an IPsec tunnel.

As an example, consider Figure 2-3. Routers R1 and R2 need to form an Open Shortest Path First (OSPF) neighborship across the service provider's cloud. Additionally, traffic between these two routers needs to be protected. While IPsec can protect unicast IP traffic, OSPF communicates through IP multicasts. Therefore, all traffic between Routers R1 and R2 (including the OSPF multicasts) is encapsulated inside of a GRE tunnel. Those GRE packets, which are unicast IP packets, are then sent across, and protected by, an IPsec tunnel.

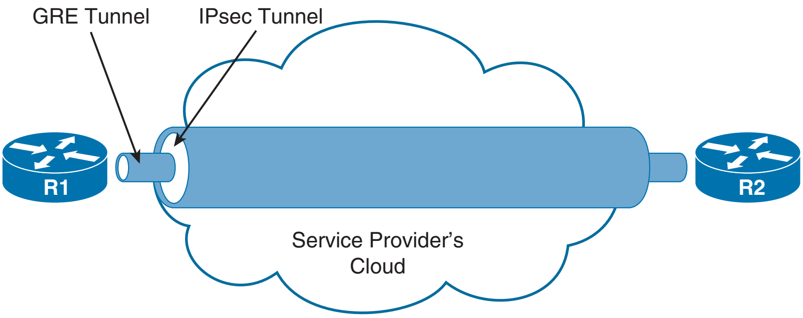

Figure 2-3 *GRE over IPsec Tunnel*

> **Note** For exam purposes, the only type of tunnel you need to know how to configure, based on the objectives listed in the ROUTE exam blueprint, is a GRE tunnel. Therefore, this chapter only provides a configuration example for a GRE tunnel.

The steps to configure a GRE tunnel are as follows:

Key Topic

Step 1. Create a virtual tunnel interface in global configuration mode with the **interface tunnel** *id* command.

Step 2. In interface configuration mode for the tunnel interface, add an IP address with the **ip address** *ip_address subnet_mask* command.

Step 3. Specify the source of the tunnel with the **tunnel source** {*interface_id* | *ip_address*} command.

Step 4. Specify the destination of the tunnel with the **tunnel destination** *ip_address* command.

Step 5. Repeat the previous steps on the router at the far side of the tunnel.

To illustrate this configuration procedure, consider Example 2-1 and the topology shown in Figure 2-4.

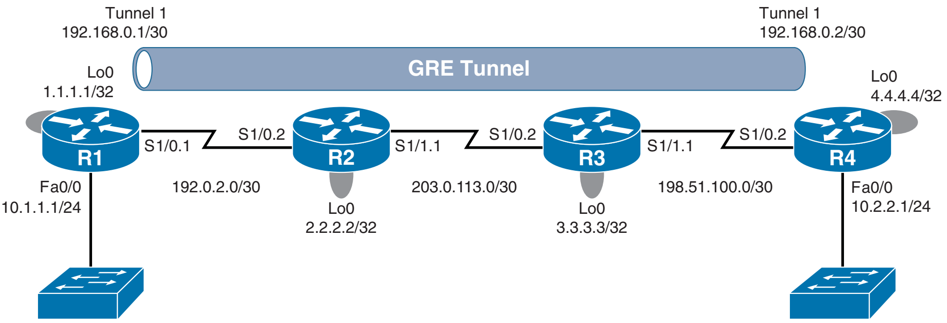

Figure 2-4 *GRE Sample Topology*

Key Topic

Example 2-1 *GRE Sample Configuration*

```
!ROUTER R1
interface Tunnel1
 ip address 192.168.0.1 255.255.255.252
 tunnel source Loopback0
 tunnel destination 4.4.4.4
```

```
!ROUTER R4
interface Tunnel1
 ip address 192.168.0.2 255.255.255.252
 tunnel source Loopback0
 tunnel destination 1.1.1.1
```

In Example 2-1, a virtual tunnel interface is created on Router R1 with the **interface Tunnel 1** command. An IP address is then assigned with the **ip address 192.168.0.1 255.255.255.252** command. Next, the **tunnel source Loopback0** command is used to specify Router R1's Lo 0 interface (and therefore its IP address of 1.1.1.1) as one end of the GRE tunnel. The **tunnel destination 4.4.4.4** command is then used to specify the Lo 0 interface on Router R4 as the other end of the tunnel. A mirrored configuration of the tunnel interface is then entered on Router R4.

Example 2-2 shows verification of the GRE tunnel. In the output of the **show interfaces tunnel 1** command, notice that the interface is up at Layer 1 and Layer 2. Also, note that the encapsulation type is TUNNEL. Also, the output of the **traceroute 192.168.0.2** command shows that the IP address of 192.168.0.2 is logically a single hop away from Router R1, even though it is physically three hops away.