

H3C 入侵防御系统产品

MCE 配置指导

新华三技术有限公司
<http://www.h3c.com>

资料版本：5W103-20180928

产品版本：T9006/T9010/T9014：E9104

T5010/T5020/ T1020/T1030/T1050/T1060/T1080：E8504

LSWM1IPSD0：E8502

Copyright © 2016-2018 新华三技术有限公司及其许可者 版权所有，保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部，并不得以任何形式传播。

除新华三技术有限公司的商标外，本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

前言

H3C IPS（Intrusion Prevention System，入侵防御系统）产品配置指导介绍了 IPS 产品各软件特性的原理及其配置方法，包含原理简介、配置任务描述和配置举例。《MCE 配置指导》主要介绍 MCE 相关的特性。

前言部分包含如下内容：

- [读者对象](#)
- [本书约定](#)
- [资料意见反馈](#)

读者对象

本手册主要适用于如下工程师：

- 网络规划人员
- 现场技术支持与维护人员
- 负责网络配置和维护的网络管理员

本书约定

1. 命令行格式约定

格 式	意 义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用 加粗 字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用 <i>斜体</i> 表示。
[]	表示用 “[]” 括起来的部分在命令配置时是可选的。
{ x y ... }	表示从多个选项中仅选取一个。
[x y ...]	表示从多个选项中选取一个或者不选。
{ x y ... } *	表示从多个选项中至少选取一个。
[x y ...] *	表示从多个选项中选取一个、多个或者不选。
&<1-n>	表示符号&前面的参数可以重复输入1~n次。
#	由“#”号开始的行表示为注释行。






2. 图形界面格式约定

格 式	意 义
<>	带尖括号“<>”表示按钮名，如“单击<确定>按钮”。
[]	带方括号“[]”表示窗口名、菜单名和数据表，如“弹出[新建用户]窗口”。
/	多级菜单用“/”隔开。如[文件/新建/文件夹]多级菜单表示[文件]菜单下的[新建]子菜单下

格 式	意 义
	的[文件夹]菜单项。

3. 各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志的意义如下：

 警告	该标志后的注释需给予格外关注，不当的操作可能会对人身造成伤害。
 注意	提醒操作中应注意的事项，不当的操作可能会导致数据丢失或者设备损坏。
 提示	为确保设备配置成功或者正常工作而需要特别关注的操作或信息。
 说明	对操作内容的描述进行必要的补充和说明。
 窍门	配置、操作、或使用设备的技巧、小窍门。

4. 图标约定

本书使用的图标及其含义如下：

	该图标及其相关描述文字代表一般网络设备，如路由器、交换机、防火墙等。
	该图标及其相关描述文字代表一般意义下的路由器，以及其他运行了路由协议的设备。
	该图标及其相关描述文字代表二、三层以太网交换机，以及运行了二层协议的设备。
	该图标及其相关描述文字代表无线控制器、无线控制器业务板和有线无线一体化交换机的无线控制引擎设备。
	该图标及其相关描述文字代表无线接入点设备。
	该图标及其相关描述文字代表无线终结单元。
	该图标及其相关描述文字代表无线终结者。
	该图标及其相关描述文字代表无线Mesh设备。
	该图标代表发散的无线射频信号。
	该图标代表点到点的无线射频信号。
	该图标及其相关描述文字代表防火墙、UTM、多业务安全网关、负载均衡等安全设备。



该图标及其相关描述文字代表防火墙插卡、负载均衡插卡、NetStream插卡、SSL VPN插卡、IPS插卡、ACG插卡等安全插卡。

5. 示例约定

由于设备型号不同、配置不同、版本升级等原因，可能造成本手册中的内容与用户使用的设备显示信息不一致。实际使用中请以设备显示的内容为准。

本手册中出现的端口编号仅作示例，并不代表设备上实际具有此编号的端口，实际使用中请以设备上存在的端口编号为准。

资料意见反馈

如果您在使用过程中发现产品资料的任何问题，可以通过以下方式反馈：

E-mail: info@h3c.com

感谢您的反馈，让我们做得更好！

目 录

1 MCE	1-1
1.1 MPLS L3VPN简介	1-1
1.1.1 MPLS L3VPN基本网络架构	1-1
1.1.2 MPLS L3VPN基本概念	1-1
1.2 MCE简介	1-3
1.3 MCE配置任务简介	1-4
1.4 配置VPN实例	1-5
1.4.1 创建VPN实例	1-5
1.4.2 配置VPN实例与接口关联	1-5
1.4.3 配置VPN实例的路由相关属性	1-6
1.5 配置MCE	1-6
1.5.1 配置MCE与站点之间的路由交换	1-7
1.5.2 配置MCE与PE之间的路由交换	1-8
1.6 MCE显示和维护	1-9
1.7 MCE典型配置举例	1-10
1.7.1 配置MCE示例	1-10
2 IPv6 MCE	2-1
2.1 IPv6 MPLS L3VPN简介	2-1
2.1.1 IPv6 MPLS L3VPN概述	2-1
2.2 IPv6 MCE简介	2-1
2.3 IPv6 MCE配置任务简介	2-1
2.4 配置VPN实例	2-2
2.4.1 创建VPN实例	2-2
2.4.2 配置VPN实例与接口关联	2-3
2.4.3 配置VPN实例的路由相关属性	2-3
2.5 配置IPv6 MCE	2-4
2.5.1 配置MCE与站点之间的路由交换	2-4
2.5.2 配置MCE与PE之间的路由交换	2-6
2.6 IPv6 MCE显示和维护	2-7
2.7 IPv6 MCE典型配置举例	2-7

1 MCE

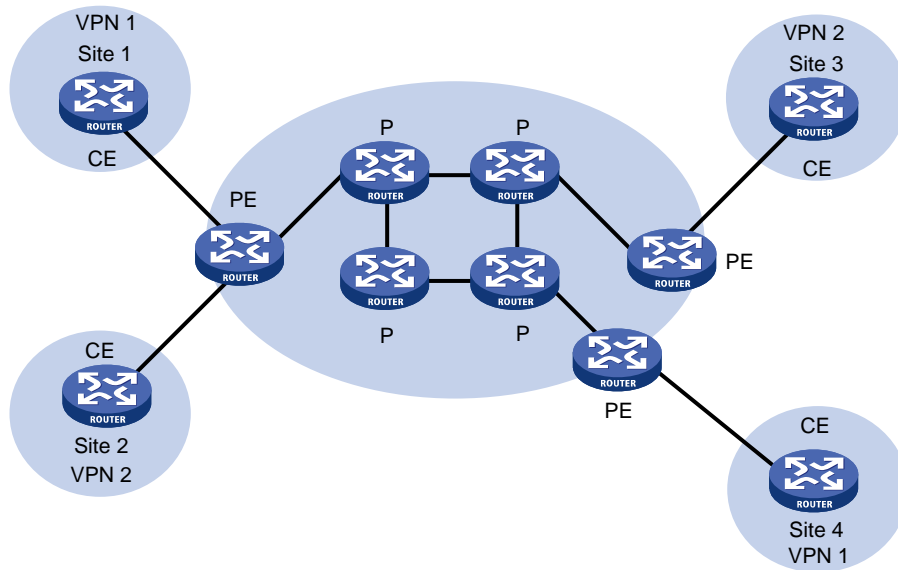
1.1 MPLS L3VPN简介

MPLS L3VPN 是一种三层 VPN 技术，它使用 BGP 在服务提供商骨干网上发布用户站点的私网路由，使用 MPLS 在服务提供商骨干网上转发用户站点之间的私网报文，从而实现通过服务提供商的骨干网连接属于同一个 VPN、位于不同地理位置的用户站点。

MPLS L3VPN 组网方式灵活，可扩展性好，并能够方便地支持 MPLS QoS 和 MPLS TE，因此得到了广泛的应用。

1.1.1 MPLS L3VPN基本网络架构

图1-1 MPLS L3VPN 基本网络架构



MPLS L3VPN的基本网络架构如 [图 1-1](#) 所示。MPLS L3VPN网络中设备的角色分为以下几种：

- CE (Customer Edge, 用户网络边缘) 设备：直接与服务提供商网络相连的用户网络侧设备。CE “感知”不到 VPN 的存在，也不需要支持 MPLS。
- PE (Provider Edge, 服务提供商网络边缘) 设备：与 CE 相连的服务提供商网络侧设备。在 MPLS L3VPN 网络中，对 VPN 的所有处理都发生在 PE 上。
- P (Provider, 服务提供商网络) 设备：服务提供商网络中的骨干设备，不与 CE 直接相连。P 只需要在骨干网中将用户网络报文转发给正确的远端 PE，不需要维护和处理 VPN 信息。

1.1.2 MPLS L3VPN基本概念

1. Site

Site (站点) 的含义可以从下述几个方面理解：

- Site 是指相互之间具备 IP 连通性的一组 IP 系统，并且这组 IP 系统的 IP 连通性不需通过服务提供商网络实现；
 - Site 的划分是根据设备的拓扑关系，而不是地理位置，尽管在大多数情况下一个 Site 中的设备地理位置相邻；
 - 一个 Site 中的设备可以属于多个 VPN，换言之，一个 Site 可以属于多个 VPN；
 - Site 通过 CE 连接到服务提供商网络，一个 Site 可以包含多个 CE，但一个 CE 只属于一个 Site。
- 对于多个连接到同一服务提供商网络的 Site，通过制定策略，可以将它们划分为不同的集合 (set)，只有属于相同集合的 Sites 之间才能通过服务提供商网络互访，这种集合就是 VPN。

2. VPN实例

在 MPLS L3VPN 中，不同 VPN 之间的路由隔离通过 VPN 实例 (VPN-instance) 实现，VPN 实例又称为 VRF (Virtual Routing and Forwarding，虚拟路由和转发) 实例。PE 上每个 VPN 实例都有相对独立的路由表和 LFIB (Label Forwarding Information Base，标签转发信息库)，确保 VPN 数据的独立性和安全性。

PE 通过将 Site 连接的接口与 VPN 实例关联，实现该 Site 与 VPN 实例的关联。一个 Site 只能与一个 VPN 实例关联；不同的 Site 可以关联同一个 VPN 实例。VPN 实例中包含了与其关联的 Site 所属的所有 VPN 的成员关系和路由规则等信息。

VPN 实例中的信息包括：LFIB、IP 路由表、与 VPN 实例关联的接口以及 VPN 实例的管理信息。VPN 实例的管理信息包括 RD (Route Distinguisher，路由标识符)、VPN Target 属性、路由过滤策略等。

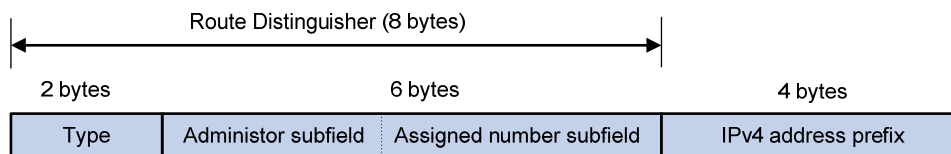
3. VPN-IPv4 地址

VPN 是一种私有网络，不同的 VPN 独立管理自己使用的地址范围，也称为地址空间 (Address Space)。不同 VPN 的地址空间可能会在一定范围内重合，比如，VPN 1 和 VPN 2 都使用了 10.110.10.0/24 网段的地址，这就发生了地址空间重叠 (Overlapping Address Spaces)。

传统 BGP 无法正确处理地址空间重叠的 VPN 的路由。假设 VPN 1 和 VPN 2 都使用了 10.110.10.0/24 网段的地址，并各自发布了一条去往此网段的路由，BGP 只会选择其中一条路由，从而导致去往另一个 VPN 的路由丢失。

MPLS L3VPN 使用 VPN-IPv4 地址 (又称为 VPNv4 地址) 来解决上述问题。

图1-2 VPN-IPv4 地址结构



如 图 1-2 所示，VPN-IPv4 地址共有 12 个字节，包括 8 字节的 RD 和 4 字节的 IPv4 地址前缀。其中，RD 的作用是将 IP 地址添加到一个 IPv4 地址前缀前，使之成为全局唯一的 VPN-IPv4 地址前缀。PE 从 CE 接收到普通 IPv4 路由后，为 IPv4 地址前缀添加 RD，将其转变为 VPN-IPv4 路由，并使用 MP-BGP (Multiprotocol Border Gateway Protocol，多协议边界网关协议) 将 VPN-IPv4 路由发布给对端 PE，从而实现通过 RD 区分不同 VPN 的相同 IPv4 地址前缀。

RD 有三种格式，通过 2 字节的 Type 字段区分：

- Type 为 0 时，Administrator 子字段占 2 字节，Assigned number 子字段占 4 字节，格式为：16 位自治系统号:32 位用户自定义数字，例如：100:1。
- Type 为 1 时，Administrator 子字段占 4 字节，Assigned number 子字段占 2 字节，格式为：32 位 IPv4 地址:16 位用户自定义数字，例如：172.1.1.1:1。
- Type 为 2 时，Administrator 子字段占 4 字节，Assigned number 子字段占 2 字节，格式为：32 位自治系统号:16 位用户自定义数字，其中的自治系统号最小值为 65536，例如：65536:1。

为了保证 VPN-IPv4 地址全球唯一，建议不要将 Administrator 子字段的值设置为私有 AS 号或私有 IP 地址。

4. VPN Target属性

MPLS L3VPN 使用 BGP 扩展团体属性——VPN Target（也称为 Route Target）来控制 VPN 路由信息的发布。

VPN Target 属性分为如下两类：

- Export Target 属性：本地 PE 从与自己直接相连的 Site 学习到 IPv4 路由后，将其转换为 VPN-IPv4 路由，为 VPN-IPv4 路由设置 Export Target 属性并发布给其它 PE。
- Import Target 属性：PE 在接收到其它 PE 发布的 VPN-IPv4 路由时，检查其 Export Target 属性。只有当此属性与 PE 上某个 VPN 实例的 Import Target 属性匹配时，才把路由加入到该 VPN 实例的路由表中。

VPN Target 属性定义了一条 VPN-IPv4 路由可以为哪些 Site 所接收，PE 可以接收哪些 Site 发送来的路由。

与 RD 类似，VPN Target 也有三种格式：

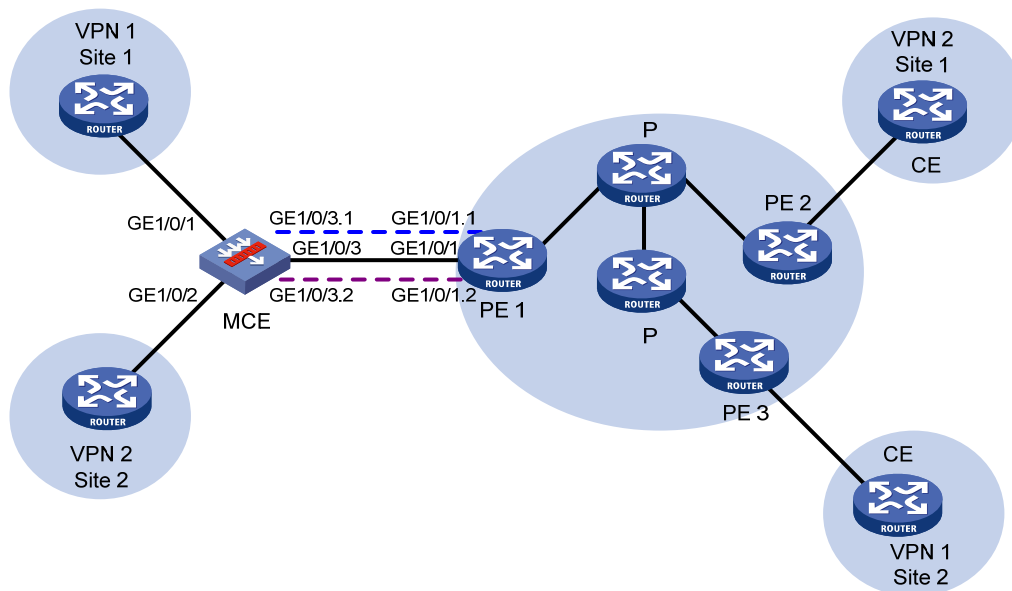
- 16 位自治系统号:32 位用户自定义数字，例如：100:1。
- 32 位 IPv4 地址:16 位用户自定义数字，例如：172.1.1.1:1。
- 32 位自治系统号:16 位用户自定义数字，其中的自治系统号最小值为 65536，例如：65536:1。

1.2 MCE简介

传统的MPLS L3VPN架构要求每个用户站点单独使用一个CE与PE相连，如 [图 1-1](#)所示。随着用户业务的不断细化和安全需求的提高，一个私有网络内的用户可能需要划分成多个VPN，不同VPN用户间的业务需要完全隔离。此时，为每个VPN单独配置一台CE将加大用户的设备开支和维护成本；而多个VPN共用一台CE，使用同一个路由表项，又无法保证数据的安全性。

MCE（Multi-VPN-Instance Customer Edge，多 VPN 实例用户网络边界设备）功能通过在 CE 设备上建立 VPN 实例，为不同的 VPN 提供逻辑独立的路由转发表和地址空间，使多个 VPN 可以共享一个 CE。该 CE 设备称为 MCE 设备。MCE 功能有效地解决了多 VPN 网络带来的用户数据安全与网络成本之间的矛盾。

图1-3 MCE 工作原理示意图



如 图 1-3 所示，MCE 组网的关键是在 MCE 与用户站点之间、MCE 与 PE 之间交互私网路由，并将其正确学习到相应 VPN 实例的路由表中。其他处理与传统的 MPLS L3VPN 相同，此处不再赘述。

- MCE 与用户站点之间的私网路由交互：在 MCE 设备上为 VPN 1 和 VPN 2 创建 VPN 实例，并使用 GigabitEthernet1/0/1 接口与 VPN 1 进行绑定、GigabitEthernet1/0/2 接口与 VPN 2 进行绑定。在接收路由信息时，MCE 设备根据路由的接收接口，即可判断该路由信息的来源，并将其维护到对应 VPN 实例的路由表中。
- MCE 与 PE 之间的私网路由交互：MCE 和 PE 1 之间通过以太网子接口连接。在 MCE 上将接口 GigabitEthernet1/0/3.1 与 VPN 1 绑定；将 GigabitEthernet1/0/3.2 与 VPN 2 绑定。在 PE 1 上为 VPN 1 和 VPN 2 创建 VPN 实例，并将连接 MCE 的接口 GigabitEthernet1/0/1.1 和 GigabitEthernet1/0/1.2 与 VPN 实例绑定，绑定的方式与 MCE 设备一致。从而，使得 MCE 与 PE 之间交互的私网路由可以准确地学习到对应 VPN 实例的路由表中。

MCE 与 VPN 站点之间、MCE 与 PE 之间可以使用静态路由、RIP 或 OSPF 交换路由信息。

1.3 MCE配置任务简介

配置 MCE 的关键在于管理 VPN 私网路由在 MPLS 骨干网上的发布，包括 MCE 与站点之间的路由交换以及 MCE 与 PE 之间的路由交换。

表1-1 MCE 配置任务简介

配置任务	说明	详细配置
配置VPN实例	创建VPN实例	必选 1.4.1
	配置VPN实例与接口关联	必选 1.4.2
	配置VPN实例的路由相关属性	可选 1.4.3
配置MCE的路	配置MCE与站点之间的路由交换	必选 1.5.1

配置任务		说明	详细配置
由交换	配置MCE与PE之间的路由交换	必选	1.5.2

1.4 配置VPN实例

VPN 实例不仅可以将 VPN 私网路由与公网路由隔离，还可以隔离不同 VPN 实例的路由，这一特点使得 VPN 实例的使用不限于 MPLS L3VPN。

配置 VPN 实例的操作是在 PE 或 MCE 设备上进行的。

1.4.1 创建VPN实例

VPN 实例在实现中与 Site 关联。VPN 实例不是直接对应于 VPN，一个 VPN 实例综合了和它所对应 Site 的 VPN 成员关系和路由规则。

表1-2 创建 VPN 实例

操作	命令	说明
进入系统视图	system-view	-
创建VPN实例，并进入VPN实例视图	ip vpn-instance <i>vpn-instance-name</i>	缺省情况下，不存在VPN实例
配置VPN实例的RD	route-distinguisher <i>route-distinguisher</i>	缺省情况下，未配置VPN实例的RD
(可选) 配置VPN实例的描述信息	description <i>text</i>	缺省情况下，未配置VPN实例的描述信息
(可选) 配置VPN实例的ID	vpn-id <i>vpn-id</i>	缺省情况下，未配置VPN实例的ID
(可选)配置VPN实例的SNMP上下文	snmp context-name <i>context-name</i>	缺省情况下，未配置VPN实例的SNMP上下文

1.4.2 配置VPN实例与接口关联



提示

执行 **ip binding vpn-instance** 命令将删除接口上已经配置的 IP 地址，因此需要重新配置接口的 IP 地址。

VPN 实例配置完成后，还需要与连接 CE 的接口进行关联。

表1-3 配置 VPN 实例与接口关联

操作	命令	说明
进入系统视图	system-view	-
进入接口视图	interface <i>interface-type interface-number</i>	-

操作	命令	说明
配置接口与指定VPN实例关联	ip binding vpn-instance <i>vpn-instance-name</i>	缺省情况下，接口未关联VPN实例，接口属于公网

1.4.3 配置VPN实例的路由相关属性

表1-4 配置 VPN 实例的路由相关属性

操作	命令	说明
进入系统视图	system-view	-
进入VPN实例视图或IPv4 VPN视图	ip vpn-instance <i>vpn-instance-name</i>	VPN实例视图下配置的路由相关属性既可以用于IPv4 VPN，也可以用于IPv6 VPN
	address-family ipv4	既可以在VPN实例视图下，也可以在IPv4 VPN视图下，配置IPv4 VPN的路由相关属性。如果同时在两个视图下配置了路由相关属性，则IPv4 VPN采用IPv4 VPN视图下配置的路由相关属性
配置VPN实例的VPN Target	vpn-target <i>vpn-target</i> <1-8> [both export-extcommunity import-extcommunity]	缺省情况下，未配置VPN实例的VPN Target
配置VPN实例支持的最大激活路由前缀数	routing-table limit <i>number</i> { <i>warn-threshold</i> simply-alert }	本命令的缺省情况与设备的型号有关，请参见命令手册描述
对当前VPN实例应用入方向路由策略	import route-policy <i>route-policy</i>	缺省情况下，允许所有VPN Target 属性匹配的路由通过 执行本配置时，需要创建路由策略。路由策略的详细介绍，请参见“三层技术-IP路由配置指导”中的“路由策略”
对当前VPN实例应用出方向路由策略	export route-policy <i>route-policy</i>	缺省情况下，不对发布的路由进行过滤 执行本配置时，需要创建路由策略。路由策略的详细介绍，请参见“三层技术-IP路由配置指导”中的“路由策略”

1.5 配置MCE

MCE 可以看作一种通过路由隔离实现业务隔离的组网方案。配置 MCE 的关键为：

- 配置 MCE 与站点之间的路由交换
- 配置 MCE 与 PE 之间的路由交换

在 MCE 组网方案中，路由计算时需要关闭 PE 上的路由环路检测功能，防止路由丢失；同时禁止各路由协议互操作功能，以节省系统资源。

在配置 MCE 之前，需要先在 MCE 上创建 VPN 实例，并将 MCE 连接站点和 PE 的接口与 VPN 实例绑定。

1.5.1 配置MCE与站点之间的路由交换

1. 配置MCE与站点之间使用静态路由

MCE 可以通过静态路由与 Site 连接。传统 CE 配置的静态路由对全局生效，无法解决多 VPN 间的地址重叠问题。MCE 功能可以将静态路由与 VPN 实例相绑定，将各 VPN 之间的静态路由进行隔离。

表1-5 配置 MCE 与站点之间使用静态路由

操作	命令	说明
进入系统视图	system-view	-
为指定VPN实例配置静态路由	ip route-static vpn-instance s-vpn-instance-name dest-address { mask-length mask } { interface-type interface-number [next-hop-address] next-hop-address [public] [track track-entry-number] vpn-instance d-vpn-instance-name next-hop-address [track track-entry-number] [permanent] [preference preference] [tag tag-value] [description text]	缺省情况下，未配置静态路由该配置在MCE上进行，站点上的配置方法与普通静态路由相同
(可选)配置静态路由的缺省优先级的	ip route-static default-preference default-preference	缺省情况下，静态路由的缺省优先级为60

2. 配置MCE与站点之间使用RIP

一个 RIP 进程只能属于一个 VPN 实例。如果在启动 RIP 进程时不绑定到 VPN 实例，则该进程属于公网进程。通过在 MCE 上将 RIP 进程与 VPN 实例绑定，可以使不同 VPN 内的私网路由通过不同的 RIP 进程在 Site 和 MCE 间进行交互，保证了私网路由的隔离和安全。RIP 的介绍和详细配置，请参见“三层技术-IP 路由配置指导”中的“RIP”。

表1-6 配置 MCE 与站点之间使用 RIP

操作	命令	说明
进入系统视图	system-view	-
创建MCE与站点间的RIP实例，并进入RIP视图	rip [process-id] vpn-instance vpn-instance-name	该配置在MCE上进行，站点上配置普通RIP即可
在指定网段接口上使能RIP	network network-address	缺省情况下，接口上的RIP功能处于关闭状态
引入由PE发布的远端站点的路由	import-route protocol [process-id all-processes] [allow-direct cost cost-value route-policy route-policy-name tag tag] *	缺省情况下，RIP未引入其它路由

3. 配置MCE与站点之间使用OSPF

一个 OSPF 进程只能属于一个 VPN 实例。如果在启动 OSPF 进程时不绑定到 VPN 实例，则该进程属于公网进程。

通过在 MCE 上将 OSPF 进程与 VPN 实例绑定，可以使不同 VPN 内的私网路由通过不同的 OSPF 进程在 Site 和 MCE 间进行交互，保证了私网路由的隔离和安全。

OSPF 的介绍和详细配置，请参见“三层技术-IP 路由配置指导”中的“OSPF”。

表1-7 配置 MCE 与站点之间使用 OSPF

操作	命令	说明
进入系统视图	system-view	-
创建MCE与站点间的OSPF实例，并进入OSPF视图	ospf [process-id router-id router-id vpn-instance vpn-instance-name] *	该配置在MCE上进行，站点上配置普通OSPF即可 VPN实例绑定的OSPF进程中不使用系统视图下配置的公网Router ID，因此用户需要在启动进程时手工配置Router ID 一个OSPF进程只能属于一个VPN实例，但一个VPN实例可以使用多个OSPF进程为其传播私网路由
引入由PE发布的远端站点的路由	import-route protocol [process-id all-processes] [allow-direct cost cost-value nssa-only route-policy route-policy-name tag tag type type] *	缺省情况下，没有引入其他协议的路由信息
配置OSPF区域，进入OSPF区域视图	area area-id	缺省情况下，未配置OSPF区域
配置区域所包含的网段并在指定网段的接口上使能OSPF	network ip-address wildcard-mask	缺省情况下，接口不属于任何区域且OSPF功能处于关闭状态

1.5.2 配置MCE与PE之间的路由交换

由于在 MCE 设备上已经将站点内的私网路由信息与 VPN 实例进行了绑定，因此，只需要在 MCE 与 PE 之间将接口与 VPN 实例进行绑定、进行简单的路由配置、并将 MCE 上维护的站点内的 VPN 路由引入到 MCE-PE 间的路由协议中，便可以实现私网 VPN 路由信息的传播。

本节中的配置均在 MCE 上进行，PE 上的配置请参见 PE 设备配套手册。

1. 配置MCE与PE之间使用静态路由

表1-8 配置 MCE 与 PE 之间使用静态路由

操作	命令	说明
进入系统视图	system-view	-
为指定VPN实例配置静态路由	ip route-static vpn-instance s-vpn-instance-name dest-address { mask-length mask } { interface-type interface-number [next-hop-address] next-hop-address [public] [track track-entry-number] vpn-instance d-vpn-instance-name next-hop-address [track track-entry-number] } [permanent] [preference preference] [tag tag-value] [description text]	缺省情况下，不存在静态路由
(可选)配置静态路由的缺省优先级	ip route-static default-preference default-preference	缺省情况下，静态路由的缺省优先级为60

2. 配置MCE与PE之间使用RIP

表1-9 配置 MCE 与 PE 之间使用 RIP

操作	命令	说明
进入系统视图	system-view	-
创建MCE与PE间的RIP实例，并进入RIP视图	rip [<i>process-id</i>] vpn-instance <i>vpn-instance-name</i>	-
在指定网段接口上使能RIP	network <i>network-address</i>	缺省情况下，接口上的RIP功能处于关闭状态
引入站点内的VPN路由	import-route <i>protocol</i> [<i>process-id</i> all-processes] [allow-direct cost <i>cost-value</i> route-policy <i>route-policy-name</i> tag <i>tag</i>] *	缺省情况下，RIP未引入其它路由

3. 配置MCE与PE之间使用OSPF

表1-10 配置 MCE 与 PE 之间使用 OSPF

操作	命令	说明
进入系统视图	system-view	-
创建MCE与PE间的OSPF实例，并进入OSPF视图	ospf [<i>process-id</i> router-id <i>router-id</i> vpn-instance <i>vpn-instance-name</i>] *	-
关闭OSPF实例的路由环路检测功能	vpn-instance-capability simple	缺省情况下，OSPF实例的路由环路检测功能处于开启状态 需要在MCE上通过本命令关闭OSPF实例的路由环路检测功能。否则，MCE不会接收PE发送过来的OSPF路由，导致路由丢失
引入站点内的VPN路由	import-route <i>protocol</i> [<i>process-id</i> all-processes] [allow-direct cost <i>cost-value</i> nssa-only route-policy <i>route-policy-name</i> tag <i>tag</i> type <i>type</i>] *	缺省情况下，没有引入其他协议的路由信息
配置OSPF区域，进入OSPF区域视图	area <i>area-id</i>	缺省情况下，未配置OSPF区域
配置区域所包含的网段并在指定网段的接口上使能OSPF	network <i>ip-address wildcard-mask</i>	缺省情况下，接口不属于任何区域且OSPF功能处于关闭状态

1.6 MCE显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 MCE 的运行情况，通过查看显示信息验证配置的效果。

表1-11 MCE 显示和维护

操作	命令
显示指定VPN实例信息	<code>display ip vpn-instance [instance-name vpn-instance-name]</code>



说明

VPN 实例中路由表的命令请参见“三层技术-IP 路由命令参考”中的“IP 路由基础命令”。

1.7 MCE 典型配置举例

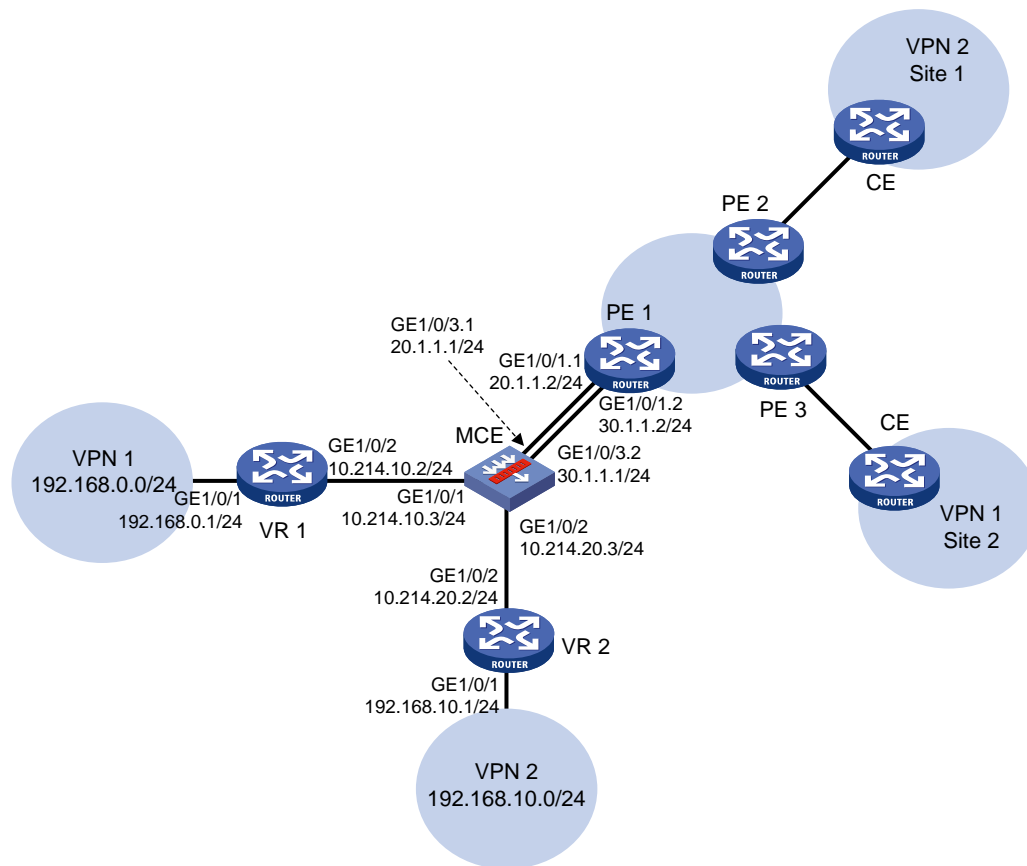
1.7.1 配置MCE示例

1. 组网需求

MCE 设备连接两个 VPN：VPN 1 和 VPN 2。其中，VPN 2 内运行 RIP 路由协议。MCE 设备将两个 VPN 之间的路由隔离，并通过 OSPF 将各 VPN 的路由发布到 PE 1。

2. 组网图

图1-4 配置 MCE 组网图



3. 配置步骤

为区分设备, 假设 MCE 系统名为“MCE”, VPN 1 和 VPN 2 的边缘设备分别名为“VR1”和“VR2”, PE 1 系统名为 “PE1”。

(1) 在 MCE 和 PE 1 上配置 VPN 实例

在 MCE 设备上配置 VPN 实例, 名称分别为 vpn1 和 vpn2, RD 分别取值为 10:1 和 20:1, VPN Target 取值与 RD 取相同数值, Export 和 Import 均取此值。

```
<MCE> system-view
[MCE] ip vpn-instance vpn1
[MCE-vpn-instance-vpn1] route-distinguisher 10:1
[MCE-vpn-instance-vpn1] vpn-target 10:1
[MCE-vpn-instance-vpn1] quit
[MCE] ip vpn-instance vpn2
[MCE-vpn-instance-vpn2] route-distinguisher 20:1
[MCE-vpn-instance-vpn2] vpn-target 20:1
[MCE-vpn-instance-vpn2] quit
```

配置接口 GigabitEthernet1/0/1 与 VPN 实例 vpn1 绑定, 并配置该接口的地址。

```
[MCE] interface gigabitethernet 1/0/1
[MCE-GigabitEthernet1/0/1] ip binding vpn-instance vpn1
[MCE-GigabitEthernet1/0/1] ip address 10.214.10.3 24
[MCE-GigabitEthernet1/0/1] quit
```

配置接口 GigabitEthernet1/0/2 与 VPN 实例 vpn2 绑定, 并配置该接口的地址。

```
[MCE] interface gigabitethernet 1/0/2
[MCE-GigabitEthernet1/0/2] ip binding vpn-instance vpn2
[MCE-GigabitEthernet1/0/2] ip address 10.214.20.3 24
[MCE-GigabitEthernet1/0/2] quit
```

在 PE 1 上配置 VPN 实例, 名称分别为 vpn1 和 vpn2, RD 分别取值为 10:1 和 20:1, VPN Target 取值与 RD 相同, Export 和 Import 均取此值。

```
<PE1> system-view
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 10:1
[PE1-vpn-instance-vpn1] vpn-target 10:1
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] route-distinguisher 20:1
[PE1-vpn-instance-vpn2] vpn-target 20:1
[PE1-vpn-instance-vpn2] quit
```

(2) MCE 与 Site 间路由配置

MCE 与 VPN 1 直接相连, 且 VPN 1 内未使用路由协议, 因此可以使用静态路由进行配置。

配置 VR1 与 MCE 连接的接口地址为 10.214.10.2/24, 连接 VPN 1 接口的地址为 192.168.0.1/24。
(具体配置过程略)

在 VR1 上配置缺省路由, 指定出方向报文的下一跳地址为 10.214.10.3。

```
<VR1> system-view
[VR1] ip route-static 0.0.0.0 0.0.0.0 10.214.10.3
```

在 MCE 上指定静态路由，去往 192.168.0.0/24 网段的报文，下一跳地址为 10.214.10.2，并将此路由与 VPN 实例 vpn1 绑定。

```
[MCE] ip route-static vpn-instance vpn1 192.168.0.0 24 10.214.10.2
```

VPN 2 内运行 RIP，在 MCE 上配置 RIP 进程 20，并与 VPN 实例 vpn2 绑定，以便将 VPN 2 内的路由学习到 VPN 实例 vpn2 的路由表中。

```
[MCE] rip 20 vpn-instance vpn2
```

发布网段 10.214.20.0 的路由。

```
[MCE-rip-20] network 10.214.20.0
```

```
[MCE-rip-20] quit
```

在 VR2 上，配置与 MCE 连接的接口地址为 10.214.20.2/24，连接 VPN 2 接口的地址为 192.168.10.1/24。（配置过程略）

配置 RIP，发布网段 192.168.10.0 和 10.214.20.0 的路由。

```
<VR2> system-view
```

```
[VR2] rip 20
```

```
[VR2-rip-20] network 192.168.10.0
```

```
[VR2-rip-20] network 10.214.20.0
```

在 MCE 上查看 VPN 实例 vpn1 和 vpn2 的路由信息。

```
[MCE] display ip routing-table vpn-instance vpn1
```

```
Destinations : 13          Routes : 13
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
10.214.10.0/24	Direct	0	0	10.214.10.3	GE1/0/1
10.214.10.0/32	Direct	0	0	10.214.10.3	GE1/0/1
10.214.10.3/32	Direct	0	0	127.0.0.1	InLoop0
10.214.10.255/32	Direct	0	0	10.214.10.3	GE1/0/1
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0
192.168.0.0/24	Static	60	0	10.214.10.2	GE1/0/1
224.0.0.0/4	Direct	0	0	0.0.0.0	NULL0
224.0.0.0/24	Direct	0	0	0.0.0.0	NULL0
255.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

```
[MCE] display ip routing-table vpn-instance vpn2
```

```
Destinations : 13          Routes : 13
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
10.214.20.0/24	Direct	0	0	10.214.20.3	GE1/0/2
10.214.20.0/32	Direct	0	0	10.214.20.3	GE1/0/2
10.214.20.3/32	Direct	0	0	127.0.0.1	InLoop0
10.214.20.255/32	Direct	0	0	10.214.20.3	GE1/0/2
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0

127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0
192.168.10.0/24	RIP	100	1	10.214.20.2	GE1/0/2
224.0.0.0/4	Direct	0	0	0.0.0.0	NULL0
224.0.0.0/24	Direct	0	0	0.0.0.0	NULL0
255.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

可以看到，MCE 已经通过 RIP 学习到了 VPN 2 内的私网路由，并与 VPN 1 内的路由信息分别维护在两个路由表内，有效地进行了隔离。

(3) MCE 与 PE 间路由配置

MCE 通过子接口与 PE 1 相连。在 MCE 上配置子接口 GigabitEthernet1/0/3.1 与 VPN 实例 vpn1 绑定，配置该子接口终结 VLAN 10，并配置该接口的地址。

```
[MCE] interface gigabitethernet 1/0/3.1
[MCE-GigabitEthernet1/0/3.1] ip binding vpn-instance vpn1
[MCE-GigabitEthernet1/0/3.1] vlan-type dot1q vid 10
[MCE-GigabitEthernet1/0/3.1] ip address 20.1.1.1 24
[MCE-GigabitEthernet1/0/3.1] quit
```

在 MCE 上配置子接口 GigabitEthernet1/0/3.2 与 VPN 实例 vpn2 绑定，配置该子接口终结 VLAN 20，并配置该接口的地址。

```
[MCE] interface gigabitethernet 1/0/3.2
[MCE-GigabitEthernet1/0/3.2] ip binding vpn-instance vpn2
[MCE-GigabitEthernet1/0/3.2] vlan-type dot1q vid 20
[MCE-GigabitEthernet1/0/3.2] ip address 30.1.1.1 24
[MCE-GigabitEthernet1/0/3.2] quit
```

在 PE 1 配置子接口 GigabitEthernet1/0/1.1 与 VPN 实例 vpn1 绑定，配置该子接口终结 VLAN 10，并配置该接口的地址。

```
[PE1] interface gigabitethernet 1/0/1.1
[PE1-GigabitEthernet1/0/1.1] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/1.1] vlan-type dot1q vid 10
[PE1-GigabitEthernet1/0/1.1] ip address 20.1.1.2 24
[PE1-GigabitEthernet1/0/1.1] quit
```

在 PE 1 上配置子接口 GigabitEthernet1/0/1.2 与 VPN 实例 vpn2 绑定，配置该子接口终结 VLAN 20，并配置该接口的地址。

```
[PE1] interface gigabitethernet 1/0/1.2
[PE1-GigabitEthernet1/0/1.2] ip binding vpn-instance vpn2
[PE1-GigabitEthernet1/0/1.2] vlan-type dot1q vid 20
[PE1-GigabitEthernet1/0/1.2] ip address 30.1.1.2 24
[PE1-GigabitEthernet1/0/1.2] quit
```

配置 MCE 和 PE 1 的 Loopback0 接口，用于指定 MCE 和 PE 1 的 Router ID，地址分别为 101.101.10.1 和 100.100.10.1。配置步骤这里省略。

配置 MCE 启动 OSPF 进程 10，该进程绑定到 VPN 实例 vpn1，关闭 OSPF 实例的路由环路检测功能，并配置域 ID 为 10。

```
[MCE] ospf 10 router-id 101.101.10.1 vpn-instance vpn1
[MCE-ospf-10] vpn-instance-capability simple
[MCE-ospf-10] domain-id 10
```

在 Area0 区域发布 20.1.1.0/24 网段路由，并引入 VPN 1 的静态路由。

```
[MCE-ospf-10] area 0
[MCE-ospf-10-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[MCE-ospf-10-area-0.0.0.0] quit
[MCE-ospf-10] import-route static
```

配置 PE 1 启动 OSPF 进程 10，绑定到 VPN 实例 vpn1，域 ID 为 10，在 Area0 区域发布 20.1.1.0/24 网段路由。

```
[PE1] ospf 10 router-id 100.100.10.1 vpn-instance vpn1
[PE1-ospf-10] domain-id 10
[PE1-ospf-10] area 0
[PE1-ospf-10-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[PE1-ospf-10-area-0.0.0.0] quit
[PE1-ospf-10] quit
```

MCE 与 PE 1 间配置 OSPF 进程 20，导入 VPN 实例 vpn2 的路由信息的过程与上面介绍的配置基本一致，不同的是在 MCE 的 OSPF 中配置导入的是 RIP 进程 20 的路由，这里不再赘述。

4. 验证配置

显示 PE 1 上的 VPN 1 路由信息。可以看到，VPN 1 内的静态路由已经引入到 MCE 与 PE 1 间的 OSPF 路由表中。

```
[PE1] display ip routing-table vpn-instance vpn1
```

```
Destinations : 13          Routes : 13

Destination/Mask    Proto  Pre Cost           NextHop             Interface
0.0.0.0/32          Direct  0  0                   127.0.0.1           InLoop0
20.1.1.0/24         Direct  0  0                   20.1.1.2            GE1/0/1.1
20.1.1.0/32         Direct  0  0                   20.1.1.2            GE1/0/1.1
20.1.1.2/32         Direct  0  0                   127.0.0.1           InLoop0
20.1.1.255/32       Direct  0  0                   20.1.1.2            GE1/0/1.1
127.0.0.0/8         Direct  0  0                   127.0.0.1           InLoop0
127.0.0.0/32       Direct  0  0                   127.0.0.1           InLoop0
127.0.0.1/32       Direct  0  0                   127.0.0.1           InLoop0
127.255.255.255/32 Direct  0  0                   127.0.0.1           InLoop0
192.168.0.0/24      O_ASE2 150 1                   20.1.1.1            GE1/0/1.1
224.0.0.0/4         Direct  0  0                   0.0.0.0             NULL0
224.0.0.0/24       Direct  0  0                   0.0.0.0             NULL0
255.255.255.255/32 Direct  0  0                   127.0.0.1           InLoop0
```

显示 PE 1 上的 VPN 2 路由信息。可以看到，VPN 2 内的 RIP 路由已经引入到 MCE 与 PE 1 间的 OSPF 路由表中。

```
[PE1] display ip routing-table vpn-instance vpn2
```

```
Destinations : 13          Routes : 13

Destination/Mask    Proto  Pre Cost           NextHop             Interface
0.0.0.0/32          Direct  0  0                   127.0.0.1           InLoop0
30.1.1.0/24         Direct  0  0                   30.1.1.2            GE1/0/1.2
30.1.1.0/32         Direct  0  0                   30.1.1.2            GE1/0/1.2
```

30.1.1.2/32	Direct	0	0	127.0.0.1	InLoop0
30.1.1.255/32	Direct	0	0	30.1.1.2	GE1/0/1.2
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0
192.168.10.0/24	O_ASE2	150	1	30.1.1.1	GE1/0/1.2
224.0.0.0/4	Direct	0	0	0.0.0.0	NULL0
224.0.0.0/24	Direct	0	0	0.0.0.0	NULL0
255.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

至此，通过配置，已经将两个 VPN 实例内的路由信息完整地传播到 PE 1 中，配置完成。

2 IPv6 MCE

2.1 IPv6 MPLS L3VPN简介

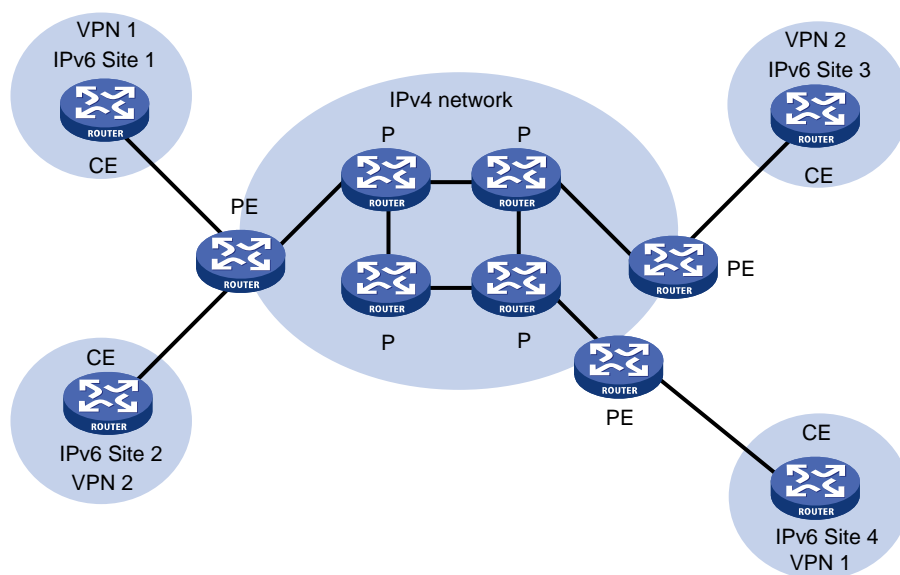
2.1.1 IPv6 MPLS L3VPN概述

MPLS L3VPN 应用于 IPv4 组网环境，利用 BGP 在服务提供商骨干网上发布 VPN 的 IPv4 路由，利用 MPLS 在服务提供商骨干网上转发 VPN 的 IPv4 报文。

IPv6 MPLS L3VPN 的原理与 MPLS L3VPN 相同，所不同的是 IPv6 MPLS L3VPN 利用 BGP 在服务提供商骨干网上发布 VPN 的 IPv6 路由，利用 MPLS 在服务提供商骨干网上转发 VPN 的 IPv6 报文。

IPv6 MPLS L3VPN 的典型组网环境如 [图 2-1](#) 所示。目前，IPv6 MPLS L3VPN 组网中服务提供商骨干网应为 IPv4 网络。VPN 内部及 CE 和 PE 之间运行 IPv6 协议，骨干网中 PE 和 P 设备之间运行 IPv4 协议。PE 需要同时支持 IPv4 和 IPv6 协议，连接 CE 的接口上使用 IPv6 协议，连接骨干网的接口上使用 IPv4 协议。

图2-1 IPv6 MPLS L3VPN 应用组网图



2.2 IPv6 MCE简介

在 IPv6 MPLS L3VPN 组网环境中，IPv6 MCE 在内网和 PE 之间发布 IPv6 路由，并交互 IPv6 报文。IPv6 MCE 的工作原理与 MCE 相同，详细内容请参见 [1.2 MCE简介](#)。

2.3 IPv6 MCE配置任务简介

配置 IPv6 MCE 的关键在于管理 VPN 私网路由在 MPLS 骨干网上的发布，包括 MCE 与站点之间的路由交换以及 MCE 与 PE 之间的路由交换。

表2-1 IPv6 MCE 配置任务简介

配置任务		说明	详细配置
配置VPN实例	创建VPN实例	必选	2.4.1
	配置VPN实例与接口关联	必选	2.4.2
	配置VPN实例的路由相关属性	可选	2.4.3
配置MCE的路由交换	配置MCE与站点之间的路由交换	必选	2.5.1
	配置MCE与PE之间的路由交换	必选	2.5.2

2.4 配置VPN实例

VPN 实例不仅可以隔离 VPN 私网路由与公网路由，还可以隔离不同 VPN 实例的路由，这一特点使得 VPN 实例的使用不限于 MPLS L3VPN。

配置 VPN 实例的操作是在 PE 或 MCE 设备上进行的。

2.4.1 创建VPN实例

VPN 实例在实现中与 Site 关联。VPN 实例不是直接对应于 VPN，一个 VPN 实例综合了和它所对应 Site 的 VPN 成员关系和路由规则。

表2-2 创建 VPN 实例

操作	命令	说明
进入系统视图	system-view	-
创建VPN实例，并进入VPN实例视图	ip vpn-instance <i>vpn-instance-name</i>	缺省情况下，不存在VPN实例
配置VPN实例的RD	route-distinguisher <i>route-distinguisher</i>	缺省情况下，未配置VPN实例的RD
(可选)配置VPN实例的描述信息	description <i>text</i>	缺省情况下，未配置VPN实例的描述信息 描述信息用于描述VPN实例，可以用来记录VPN实例与某个VPN的关系等信息
(可选)配置VPN实例的ID	vpn-id <i>vpn-id</i>	缺省情况下，未配置VPN实例的ID
(可选)配置VPN实例的SNMP上下文	snmp context-name <i>context-name</i>	缺省情况下，未配置VPN实例的SNMP上下文

2.4.2 配置VPN实例与接口关联



执行 **ip binding vpn-instance** 命令将删除接口上已经配置的 IPv6 地址，因此需要重新配置接口的 IPv6 地址。

VPN 实例配置完成后，还需要与连接 CE 的接口进行关联。

表2-3 配置 VPN 实例与接口关联

操作	命令	说明
进入系统视图	system-view	-
进入接口视图	interface <i>interface-type interface-number</i>	-
配置接口与指定VPN实例关联	ip binding vpn-instance <i>vpn-instance-name</i>	缺省情况下，接口未关联 VPN实例，接口属于公网

2.4.3 配置VPN实例的路由相关属性

表2-4 配置 VPN 实例的路由相关属性

操作	命令	说明
进入系统视图	system-view	-
进入VPN实例视图或IPv6 VPN视图	ip vpn-instance <i>vpn-instance-name</i>	VPN实例视图下配置的路由相关属性既可以用于IPv4 VPN，也可以用于IPv6 VPN
	address-family ipv6	既可以在VPN实例视图下，也可以在IPv6 VPN视图下，配置IPv6 VPN的路由相关属性。如果同时在两个视图下配置了路由相关属性，则IPv6 VPN采用IPv6 VPN视图下配置的路由相关属性
配置VPN Target	vpn-target <i>vpn-target</i> <1-8> [both export-extcommunity import-extcommunity]	缺省情况下，未配置VPN实例的VPN Target
配置支持的最大激活路由前缀数	routing-table limit <i>number</i> { <i>warn-threshold</i> simple-alert }	本命令的缺省情况请参见命令手册 配置一个VPN实例可以支持的最大激活路由前缀数，可以防止PE路由器上保存过多的激活路由前缀信息
应用入方向路由策略	import route-policy <i>route-policy</i>	缺省情况下，接收所有VPN Target属性匹配的路由 执行本配置时，需要创建路由策略。路由策略的详细介绍，请参见“三层技术-IP路由配置指导”中的“路由策略”
应用出方向路由策略	export route-policy <i>route-policy</i>	缺省情况下，不对发布的路由进行过滤 执行本配置时，需要创建路由策略。路由

操作	命令	说明
		策略的详细介绍，请参见“三层技术-IP路由配置指导”中的“路由策略”

2.5 配置IPv6 MCE

MCE 可以看作一种通过路由隔离实现业务隔离的组网方案。配置 MCE 的关键为：

- 配置 MCE 与站点之间的路由交换
- 配置 MCE 与 PE 之间的路由交换

在 MCE 组网方案中，路由计算时需要关闭 PE 上的路由环路检测功能，防止路由丢失；同时禁止各路由协议互操作功能，以节省系统资源。

在配置 MCE 之前，需要先在 MCE 上创建 VPN 实例，并将 MCE 连接站点和 PE 的接口与 VPN 实例绑定。

2.5.1 配置MCE与站点之间的路由交换

1. 配置MCE与站点之间使用IPv6 静态路由

MCE 可以通过 IPv6 静态路由与 Site 连接。传统 CE 配置的 IPv6 静态路由对全局生效，无法解决多 VPN 间的地址重叠问题。以太网交换机提供的 MCE 功能可以将 IPv6 静态路由与 VPN 实例相绑定，将各 IPv6 VPN 之间的 IPv6 静态路由进行隔离。

表2-5 配置 MCE 与站点之间使用 IPv6 静态路由

操作	命令	说明
进入系统视图	system-view	-
为指定VPN实例配置IPv6静态路由	ipv6 route-static vpn-instance s-vpn-instance-name <i>ipv6-address prefix-length { interface-type interface-number</i> <i>[next-hop-address] nexthop-address [public] vpn-instance</i> <i>d-vpn-instance-name nexthop-address } [permanent]</i> <i>[preference preference] [tag tag-value] [description text]</i>	缺省情况下，未配置 IPv6 静态路由 该配置在 MCE 上进行，站点上的配置方法与普通 IPv6 静态路由相同
(可选) 配置 IPv6 静态路由的缺省优先级	ipv6 route-static default-preference default-preference	缺省情况下，IPv6 静态路由的缺省优先级为 60

2. 配置MCE与站点之间使用RIPng

一个 RIPng 进程只能属于一个 IPv6 VPN 实例。如果在启动 RIPng 进程时不绑定到 IPv6 VPN 实例，则该进程属于公网进程。通过在 MCE 上将 RIPng 进程与 IPv6 VPN 实例绑定，可以使不同 IPv6 VPN 内的私网路由通过不同的 RIPng 进程在 Site 和 MCE 间进行交互，保证了私网路由的隔离和安全。有关 RIPng 的介绍和详细配置，请参见“三层技术-IP 路由配置指导”中的“RIPng”。

表2-6 配置 MCE 与站点之间使用 RIPng

操作	命令	说明
进入系统视图	system-view	-
创建MCE与站点间的RIPng实例，并进入RIPng视图	ripng [process-id] vpn-instance vpn-instance-name	该配置在MCE上进行，站点上配置普通RIPng即可
引入由PE发布的远端站点的路由	import-route protocol [process-id] [allow-direct cost cost-value route-policy route-policy-name] *	缺省情况下，RIPng未引入其它路由
退回系统视图	quit	-
进入接口视图	interface interface-type interface-number	-
在接口上使能RIPng路由协议	ripng process-id enable	缺省情况下，接口禁用RIPng路由协议

3. 配置MCE与站点之间使用OSPFv3

一个 OSPFv3 进程只能属于一个 IPv6 VPN 实例。如果在启动 OSPFv3 进程时不绑定到 IPv6 VPN 实例，则该进程属于公网进程。

通过在 MCE 上将 OSPFv3 进程与 IPv6 VPN 实例绑定，可以使不同 IPv6 VPN 内的私网路由通过不同的 OSPFv3 进程在 Site 和 MCE 间进行交互，保证了私网路由的隔离和安全。

有关 OSPFv3 的介绍和详细配置，请参见“三层技术-IP 路由配置指导”中的“OSPFv3”。

表2-7 配置 MCE 与站点之间使用 OSPFv3

操作	命令	说明
进入系统视图	system-view	-
创建MCE与站点间的OSPFv3实例，并进入OSPFv3视图	ospfv3 [process-id] vpn-instance vpn-instance-name] *	该配置在MCE上进行，站点上配置普通OSPFv3即可 删除VPN实例后，相关的所有OSPFv3进程也将全部被删除
配置Router ID	router-id router-id	-
引入由PE发布的远端站点的路由	import-route protocol [process-id all-processes] [allow-direct cost cost-value nssa-only route-policy route-policy-name tag tag type type] *	缺省情况下，没有引入其他协议的路由信息
退回系统视图	quit	-
进入接口视图	interface interface-type interface-number	-
在接口上使能OSPFv3	ospfv3 process-id area area-id [instance instance-id]	缺省情况下，接口上没有使能OSPFv3

2.5.2 配置MCE与PE之间的路由交换

由于在 MCE 设备上已经将站点内的私网路由信息与 IPv6 VPN 实例进行了绑定，因此，只需要在 MCE 与 PE 之间将接口与 IPv6 VPN 实例进行绑定、进行简单的路由配置、并将 MCE 上维护的站点内的 IPv6 VPN 路由引入到 MCE-PE 间的路由协议中，便可以实现私网 VPN 路由信息的传播。本节中的配置均在 MCE 上进行，PE 上的配置请参见 PE 设备配套手册。

1. 配置MCE与PE之间使用IPv6 静态路由

表2-8 配置 MCE 与 PE 之间使用 IPv6 静态路由

操作	命令	说明
进入系统视图	system-view	-
为指定VPN实例配置IPv6静态路由	ipv6 route-static vpn-instance s-vpn-instance-name <i>ipv6-address prefix-length { interface-type interface-number</i> <i>[next-hop-address] nexthop-address [public] vpn-instance</i> <i>d-vpn-instance-name nexthop-address } [permanent]</i> <i>[preference preference] [tag tag-value] [description text]</i>	缺省情况下，未配置 IPv6 静态路由
(可选) 配置IPv6静态路由的缺省优先级	ipv6 route-static default-preference default-preference	缺省情况下，IPv6 静态路由的缺省优先级为60

2. 配置MCE与PE之间使用RIPng

表2-9 配置 MCE 与 PE 之间使用 RIPng

操作	命令	说明
进入系统视图	system-view	-
创建MCE与PE间的RIPng实例，并进入RIPng视图	ripng [process-id] vpn-instance <i>vpn-instance-name</i>	-
引入站点内的VPN路由	import-route protocol [process-id] <i>[allow-direct cost cost-value route-policy</i> <i>route-policy-name] *</i>	缺省情况下，RIPng未引入其它路由
退回系统视图	quit	-
进入接口视图	interface interface-type interface-number	-
在指定的网络接口上使能RIPng	ripng process-id enable	缺省情况下，接口禁用RIPng

3. 配置MCE与PE之间使用OSPFv3

表2-10 配置 MCE 与 PE 之间使用 OSPFv3

操作	命令	说明
进入系统视图	system-view	-
创建MCE与PE间的OSPFv3实例，并进入OSPFv3视图	ospfv3 [process-id] vpn-instance <i>vpn-instance-name] *</i>	-
配置Router ID	router-id router-id	-

操作	命令	说明
关闭OSPFv3实例的路由环路检测功能	vpn-instance-capability simple	缺省情况下, OSPFv3实例的路由环路检测功能处于开启状态 需要在MCE上通过本命令关闭OSPFv3实例的路由环路检测功能。否则, MCE不会接收PE发送过来的OSPFv3路由, 导致路由丢失
引入站点内的VPN路由	import-route <i>protocol</i> [<i>process-id</i> all-processes] [allow-direct cost <i>cost-value</i> nssa-only route-policy <i>route-policy-name</i> tag <i>tag</i> type <i>type</i>] *	缺省情况下, 没有引入其他协议的路由信息
退回系统视图	quit	-
进入接口视图	interface <i>interface-type</i> <i>interface-number</i>	-
在接口上使能OSPFv3	ospfv3 <i>process-id</i> area <i>area-id</i> [instance <i>instance-id</i>]	缺省情况下, 接口上没有使能OSPFv3

2.6 IPv6 MCE显示和维护

在完成上述配置后, 在任意视图下执行 **display** 命令可以显示配置后 IPv6 MCE 的运行情况, 通过查看显示信息验证配置的效果。

表2-11 IPv6 MCE 显示和维护

操作	命令
显示指定VPN实例信息	display ip vpn-instance [instance-name <i>vpn-instance-name</i>]



说明

VPN 实例中路由表的命令请参见“三层技术-IP 路由命令参考”中的“IP 路由基础命令”。

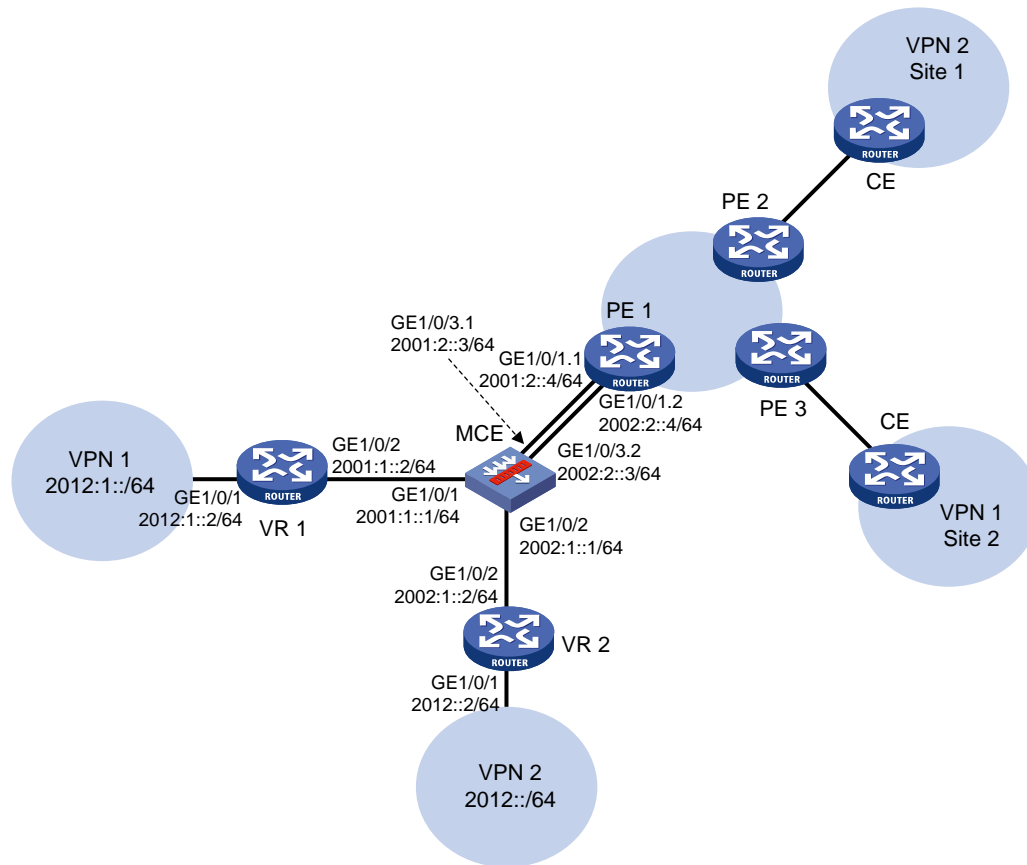
2.7 IPv6 MCE典型配置举例

1. 组网需求

MCE 设备连接两个 VPN: VPN 1 和 VPN 2, VPN 2 内运行 RIPng 路由协议。MCE 设备将两个 VPN 之间的路由隔离, 并通过 OSPFv3 将各 VPN 的路由发布到 PE 1。

2. 组网图

图2-2 配置 IPv6 MCE 组网图



3. 配置步骤

为区分设备，假设 MCE 系统名为“MCE”，VPN 1 和 VPN 2 的边缘设备分别名为“VR1”和“VR2”，PE 1 系统名为“PE1”。

(1) 在 MCE 和 PE 1 上配置 VPN 实例

在 MCE 设备上配置 VPN 实例，名称分别为 vpn1 和 vpn2，RD 分别取值为 10:1 和 20:1，VPN Target 取值与 RD 取相同数值，Export 和 Import 均取此值。

```
<MCE> system-view
[MCE] ip vpn-instance vpn1
[MCE-vpn-instance-vpn1] route-distinguisher 10:1
[MCE-vpn-instance-vpn1] vpn-target 10:1
[MCE-vpn-instance-vpn1] quit
[MCE] ip vpn-instance vpn2
[MCE-vpn-instance-vpn2] route-distinguisher 20:1
[MCE-vpn-instance-vpn2] vpn-target 20:1
[MCE-vpn-instance-vpn2] quit
```

配置接口 GigabitEthernet1/0/1 与 VPN 实例 vpn1 绑定，并配置该接口的 IPv6 地址。

```
[MCE] interface gigabitethernet 1/0/1
[MCE-GigabitEthernet1/0/1] ip binding vpn-instance vpn1
```

```
[MCE-GigabitEthernet1/0/1] ipv6 address 2001:1::1 64
[MCE-GigabitEthernet1/0/1] quit
```

配置接口 GigabitEthernet1/0/2 与 VPN 实例 vpn2 绑定，并配置该接口的 IPv6 地址。

```
[MCE] interface gigabitethernet 1/0/2
[MCE-GigabitEthernet1/0/2] ip binding vpn-instance vpn2
[MCE-GigabitEthernet1/0/2] ipv6 address 2002:1::1 64
[MCE-GigabitEthernet1/0/2] quit
```

在 PE 1 上配置 VPN 实例，名称分别为 vpn1 和 vpn2，RD 分别取值为 10:1 和 20:1，VPN Target 取值与 RD 相同，Export 和 Import 均取此值。

```
<PE1> system-view
[PE1] ip vpn-instance vpn1
[PE1-vpn-instance-vpn1] route-distinguisher 10:1
[PE1-vpn-instance-vpn1] vpn-target 10:1
[PE1-vpn-instance-vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn-instance-vpn2] route-distinguisher 20:1
[PE1-vpn-instance-vpn2] vpn-target 20:1
[PE1-vpn-instance-vpn2] quit
```

(2) MCE 与 Site 间路由配置

MCE 与 VPN 1 直接相连，且 VPN 1 内未使用路由协议，因此可以使用 IPv6 静态路由进行配置。

配置 VR1 与 MCE 连接的接口地址为 2001:1::2/64，连接 VPN 1 接口的地址为 2012:1::2/64。（具体配置过程略）

在 VR1 上配置缺省路由，指定出方向报文的下一跳地址为 2001:1::1。

```
<VR1> system-view
[VR1] ipv6 route-static :: 0 2001:1::1
```

在 MCE 上指定 IPv6 静态路由，去往 2012:1::/64 网段的报文，下一跳地址为 2001:1::2，并将此路由与 VPN 实例 vpn1 绑定。

```
[MCE] ipv6 route-static vpn-instance vpn1 2012:1:: 64 2001:1::2
```

VPN 2 内运行 RIPng，在 MCE 上配置 RIPng 进程 20，并与 VPN 实例 vpn2 绑定，以便将 VPN 2 内的路由学习到 vpn2 实例的路由表中。

```
[MCE] ripng 20 vpn-instance vpn2
```

配置 RIPng 发布 2002:1::/64 网段路由。

```
[MCE] interface gigabitethernet 1/0/2
[MCE-GigabitEthernet1/0/2] ripng 20 enable
[MCE-GigabitEthernet1/0/2] quit
```

在 VR 2 上，配置与 MCE 连接的接口地址为 2002:1::2/64。（具体配置过程略）

在 VR 2 上配置 RIPng 发布 2012::/64 和 2002:1::/64 网段路由。

```
<VR2> system-view
[VR2] ripng 20
[VR2-ripng-20] quit
[VR2] interface gigabitethernet 1/0/1
[VR2-GigabitEthernet1/0/1] ripng 20 enable
[VR2-GigabitEthernet1/0/1] quit
[VR2] interface gigabitethernet 1/0/2
[VR2-GigabitEthernet1/0/2] ripng 20 enable
```

```
[VR2-GigabitEthernet1/0/2] quit
```

```
# 在 MCE 上查看 VPN 实例 vpn1 和 vpn2 的路由信息。
```

```
[MCE] display ipv6 routing-table vpn-instance vpn1
```

```
Destinations : 6 Routes : 6
```

```
Destination: ::1/128                Protocol : Direct
NextHop    : ::1                    Preference: 0
Interface  : InLoop0                Cost      : 0
```

```
Destination: 2001:1::/64             Protocol : Direct
NextHop    : ::                     Preference: 0
Interface  : GE1/0/1                Cost      : 0
```

```
Destination: 2001:1::1/128           Protocol : Direct
NextHop    : ::1                    Preference: 0
Interface  : InLoop0                Cost      : 0
```

```
Destination: 2012:1::/64             Protocol : Static
NextHop    : 2001:1::2              Preference: 60
Interface  : GE1/0/1                Cost      : 0
```

```
Destination: FE80::/10               Protocol : Direct
NextHop    : ::                     Preference: 0
Interface  : NULL0                  Cost      : 0
```

```
Destination: FF00::/8                Protocol : Direct
NextHop    : ::                     Preference: 0
Interface  : NULL0                  Cost      : 0
```

```
[MCE] display ipv6 routing-table vpn-instance vpn2
```

```
Destinations : 6 Routes : 6
```

```
Destination: ::1/128                Protocol : Direct
NextHop    : ::1                    Preference: 0
Interface  : InLoop0                Cost      : 0
```

```
Destination: 2002:1::/64             Protocol : Direct
NextHop    : ::                     Preference: 0
Interface  : GE1/0/2                Cost      : 0
```

```
Destination: 2002:1::1/128           Protocol : Direct
NextHop    : ::1                    Preference: 0
Interface  : InLoop0                Cost      : 0
```

```
Destination: 2012::/64               Protocol : RIPng
NextHop    : FE80::20C:29FF:FE40:701 Preference: 100
Interface  : GE1/0/2                Cost      : 1
```

```
Destination: FE80::/10                Protocol : Direct
NextHop      : ::                      Preference: 0
Interface    : NULL0                   Cost      : 0
```

```
Destination: FF00::/8                Protocol : Direct
NextHop      : ::                      Preference: 0
Interface    : NULL0                   Cost      : 0
```

可以看到，MCE 已经通过 RIPng 学习到了 VPN 2 内的私网路由，并与 VPN 1 内的路由信息分别维护在两个路由表内，有效地进行了隔离。

(3) MCE 与 PE 间路由配置

MCE 通过子接口与 PE 1 相连。在 MCE 上配置子接口 GigabitEthernet1/0/3.1 与 VPN 实例 vpn1 绑定，配置该子接口终结 VLAN 10，并配置该接口的 IPv6 地址。

```
[MCE] interface gigabitethernet 1/0/3.1
[MCE-GigabitEthernet1/0/3.1] ip binding vpn-instance vpn1
[MCE-GigabitEthernet1/0/3.1] vlan-type dot1q vid 10
[MCE-GigabitEthernet1/0/3.1] ipv6 address 2001:2::3 64
[MCE-GigabitEthernet1/0/3.1] quit
```

在 MCE 上配置子接口 GigabitEthernet1/0/3.2 与 VPN 实例 vpn2 绑定，配置该子接口终结 VLAN 20，并配置该接口的 IPv6 地址。

```
[MCE] interface gigabitethernet 1/0/3.2
[MCE-GigabitEthernet1/0/3.2] ip binding vpn-instance vpn2
[MCE-GigabitEthernet1/0/3.2] vlan-type dot1q vid 20
[MCE-GigabitEthernet1/0/3.2] ipv6 address 2002:2::3 64
[MCE-GigabitEthernet1/0/3.2] quit
```

在 PE 1 配置子接口 GigabitEthernet1/0/1.1 与 VPN 实例 vpn1 绑定，配置该子接口终结 VLAN 10，并配置该接口的 IPv6 地址。

```
[PE1] interface gigabitethernet 1/0/1.1
[PE1-GigabitEthernet1/0/1.1] ip binding vpn-instance vpn1
[PE1-GigabitEthernet1/0/1.1] vlan-type dot1q vid 10
[PE1-GigabitEthernet1/0/1.1] ipv6 address 2001:2::4 64
[PE1-GigabitEthernet1/0/1.1] quit
```

在 PE 1 上配置子接口 GigabitEthernet1/0/1.2 与 VPN 实例 vpn2 绑定，配置该子接口终结 VLAN 20，并配置该接口的 IPv6 地址。

```
[PE1] interface gigabitethernet 1/0/1.2
[PE1-GigabitEthernet1/0/1.2] ip binding vpn-instance vpn2
[PE1-GigabitEthernet1/0/1.2] vlan-type dot1q vid 20
[PE1-GigabitEthernet1/0/1.2] ipv6 address 2002:2::4 64
[PE1-GigabitEthernet1/0/1.2] quit
```

配置 MCE 和 PE 1 的 Loopback0 接口，用于指定 MCE 和 PE 1 的 Router ID，地址分别为 101.101.10.1 和 100.100.10.1。配置步骤这里省略。

配置 MCE 启动 OSPFv3 进程 10，绑定到 VPN 实例 vpn1，并引入 VPN 1 的 IPv6 静态路由。

```
[MCE] ospfv3 10 vpn-instance vpn1
[MCE-ospf-10] router-id 101.101.10.1
[MCE-ospf-10] import-route static
```



```
[MCE-ospf-10] quit
```

在子接口 GigabitEthernet1/0/3.1 上使能 OSPFv3。

```
[MCE] interface gigabitethernet 1/0/3.1
```

```
[MCE-GigabitEthernet1/0/3.1] ospfv3 10 area 0.0.0.0
```

```
[MCE-GigabitEthernet1/0/3.1] quit
```

配置 PE 1 启动 OSPFv3 进程 10，绑定到 VPN 实例 vpn1。

```
[PE1] ospfv3 10 vpn-instance vpn1
```

```
[PE1-ospf-10] router-id 100.100.10.1
```

```
[PE1-ospf-10] quit
```

在子接口 GigabitEthernet1/0/1.1 上使能 OSPFv3。

```
[PE1] interface gigabitethernet 1/0/1.1
```

```
[PE1-GigabitEthernet1/0/1.1] ospfv3 10 area 0.0.0.0
```

```
[PE1-GigabitEthernet1/0/1.1] quit
```

MCE 与 PE 1 间配置 OSPFv3 进程 20，引入 VPN 实例 vpn2 的路由信息的过程与上面介绍的配置基本一致，不同的是在 MCE 的 OSPFv3 中配置引入的是 RIPng 进程 20 的路由，这里不再赘述。

4. 验证配置

显示 PE 1 上 VPN 实例 vpn1 的路由信息。可以看到，PE 1 通过 OSPFv3 学习到了 VPN 1 内的私网路由。

```
[PE1] display ipv6 routing-table vpn-instance vpn1
```

```
Destinations : 6 Routes : 6
```

```
Destination: ::1/128                Protocol : Direct
NextHop      : ::1                  Preference: 0
Interface    : InLoop0              Cost      : 0
```

```
Destination: 2001:2::/64             Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : GE1/0/1.1            Cost      : 0
```

```
Destination: 2001:2::4/128           Protocol : Direct
NextHop      : ::1                  Preference: 0
Interface    : InLoop0              Cost      : 0
```

```
Destination: 2012:1::/64             Protocol : O_ASE2
NextHop      : FE80::200:5EFF:FE01:1C05 Preference: 15
Interface    : GE1/0/1.1            Cost      : 10
```

```
Destination: FE80::/10               Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : NULL0                Cost      : 0
```

```
Destination: FF00::/8                Protocol : Direct
NextHop      : ::                  Preference: 0
Interface    : NULL0                Cost      : 0
```

通过下面的显示信息可以看出，PE 1 通过 OSPFv3 学习到了 VPN 2 内的私网路由。

```
[PE1] display ipv6 routing-table vpn-instance vpn2
```

```
Destinations : 6 Routes : 6
```

```
Destination: ::1/128          Protocol : Direct
NextHop      : ::1           Preference: 0
Interface    : InLoop0       Cost      : 0
```

```
Destination: 2002:2::/64      Protocol : Direct
NextHop      : ::            Preference: 0
Interface    : GE1/0/1.2      Cost      : 0
```

```
Destination: 2002:2::4/128    Protocol : Direct
NextHop      : ::1           Preference: 0
Interface    : InLoop0       Cost      : 0
```

```
Destination: 2012::/64        Protocol : O_ASE2
NextHop      : FE80::200:5EFF:FE01:1C06 Preference: 15
Interface    : GE1/0/1.2      Cost      : 10
```

```
Destination: FE80::/10        Protocol : Direct
NextHop      : ::            Preference: 0
Interface    : NULL0         Cost      : 0
```

```
Destination: FF00::/8         Protocol : Direct
NextHop      : ::            Preference: 0
Interface    : NULL0         Cost      : 0
```

至此，通过配置，已经将两个 VPN 实例内的路由信息完整地传播到 PE 1 中，配置完成。