

**HUAWEI NetEngine 8100 X, NetEngine 8000 X,  
NetEngine 8000E X 系列  
V800R023C00SPC500**

**配置指南**

文档版本	01
发布日期	2023-09-30



版权所有 © 华为技术有限公司 2023。保留一切权利。

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

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<https://www.huawei.com>

客户服务邮箱：[support@huawei.com](mailto:support@huawei.com)

客户服务电话：4008302118

# 目 录

<b>1 配置</b>	<b>1</b>
1.1 应用感知	1
1.1.1 应用感知配置	1
1.1.1.1 APN6 特性描述	1
1.1.1.1.1 APN6 介绍	1
1.1.1.1.2 APN6 原理描述	2
1.1.1.1.3 APN6 应用	12
1.1.1.2 APN6 配置	14
1.1.1.2.1 APN6 概述	14
1.1.1.2.2 APN6 特性限制	14
1.1.1.2.3 配置 APN6 基本功能	14
1.1.1.2.4 （可选）配置 APN ID 引流	18
1.1.1.2.5 （可选）配置 APN ID 隔离策略	19
1.1.1.2.6 维护 APN6	20
1.1.1.2.7 配置举例	21

# 插图目录

图 1-1 基于 APN6 的应用感知..... 2

图 1-2 APN6 的关键要素..... 3

图 1-3 APN6 的网络架构..... 4

图 1-4 IPv6 扩展头提供的可编程空间..... 4

图 1-5 SRH 提供的可编程空间..... 5

图 1-6 APN 标识位置..... 6

图 1-7 APN Header 的格式..... 7

图 1-8 通过 APN ID 引流..... 8

图 1-9 APN6 数据转发..... 10

图 1-10 隔离类型..... 11

图 1-11 隔离流程..... 11

图 1-12 智能云网方案..... 12

图 1-13 金融行业应用..... 13

图 1-14 配置 L3VPNv4 over SRv6 TE Flow Group 组网图..... 21

图 1-15 配置 EVPN L3VPNv6 over SRv6 TE Policy 组网图..... 40

# 表格目录

表 1-1 Option 区块字段解释.....	6
表 1-2 APN Header 各字段的含义.....	7

# 1 配置

---

## 1.1 应用感知

### 1.1.1 应用感知配置

#### 1.1.1.1 APN6 特性描述

##### 1.1.1.1.1 APN6 介绍

### 定义

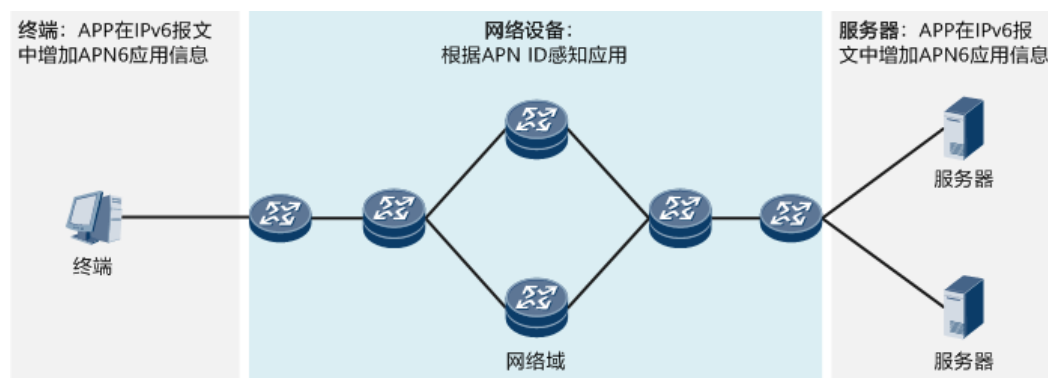
APN6（Application-aware IPv6 Networking，应用感知型IPv6网络）是一种新的网络架构，它利用IPv6报文自带的可编程空间，将应用信息（APN Attribute），包括应用标识信息（APN ID）和应用需求参数信息（APN Parameters）携带进入网络，进而为服务提供商提供精细的网络服务和精准的网络运维，当前版本应用信息中只支持携带APN ID。

### 目的

随着应用差异化需求的不断涌现，网络技术与服务也随之不断丰富。这为网络运营和运维带来了相应的挑战：如何有效实现精细网络服务、精准网络运维，APN6是满足应用差异化需求和SLA保障、促进网络持续发展与演进的关键。

如图1-1所示，APN6通过IPv6扩展头将业务报文的APN ID携带进入网络，使得网络具备感知应用的能力。

图 1-1 基于 APN6 的应用感知



基于IPv6的APN6具有如下技术优势：

- 简单直接：基于IP可达性，利用IPv6自身的报文封装直接实现应用信息的携带。
- 无缝融合：端侧/云侧和网络侧都基于IPv6，易于实现应用和网络的无缝融合。
- 扩展性强：IPv6报文封装提供的可编程空间可以用来携带丰富的应用相关信息。
- 兼容性好：如果应用信息不被网络节点识别，报文即作为普通IPv6报文被转发。
- 依赖性弱：应用信息传递和处理都基于转发平面，区别于SDN方式涉及多接口。
- 响应快速：应用信息传递和处理都基于转发平面，可以做到流驱动的直接响应。

## 受益

APN6能够有效感知关键应用（组）、关键用户（组）、及其对网络的性能需求。对于服务提供商来说，SRv6、网络切片、DetNet、SFC、SD-WAN、IFIT等技术与APN6的结合，将极大地丰富云网服务维度，扩大云网商业增值空间，使能云网精细化运营。对于终端用户来说，可以享受更加精细的网络服务。

- 敏捷的自动选路：通过APN ID精细标识关键的应用（组）或用户（组），引导进入相应的SRv6 TE Policy，网络切片，或者SFC（Service Function Chain，业务功能链）路径等，实现应用分流和灵活选路。
- 应用流量可视监控：结合APN6，人工智能和大数据分析可以对关键应用或用户进行流量特征画像，呈现其流量路径、特征、变化规律及趋势，实现应用流量可视监控。
- 智能的网络运维：通过APN ID精细标识关键的应用（组）或用户（组），实施随路检测IFIT（逐跳、端到端两种选择模式），可以实现关键业务的实时性能监控。

### 1.1.1.1.2 APN6 原理描述

## APN6 基本原理

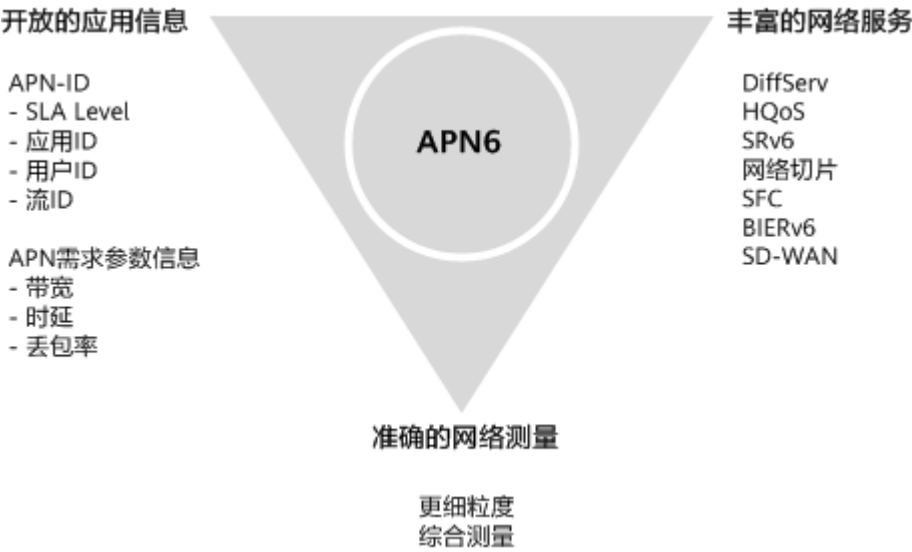
## APN6 的三要素

APN6共包含了3个关键要素，如图1-2所示。

1. 开放的应用信息：APN6通过Native IPv6报文自带的编程空间携带应用信息，包括标识信息（APN ID）和应用对网络的需求信息（APN Parameters），针对不同应用（类/组）流量进行精细化区分和网络内调度。而封装和携带应用信息的权利，则是面向应用开放的，应用可以根据需要自主选择。

- 2. 丰富的网络服务：近年来，随着网络技术不断发展，为应用提供的网络服务越来越丰富，包括SRv6 TE Policy，网络切片，SFC等，能够保障不同应用的各种差异化需求。APN6可以和这些网络新型技术结合，使能更加精细化的网络服务。例如，APN6结合网络切片技术，可以精细地为某个/组关键应用和用户提供专属网络切片，赋予专属资源，提供差异化的SLA和可靠性保障。
- 3. 准确的网络测量：网络测量技术不断丰富。APN6可以和网络测量技术相结合，提供应用级精细化可视、实时性能测量和快速故障定界等能力，使能精准网络运维。

图 1-2 APN6 的关键要素

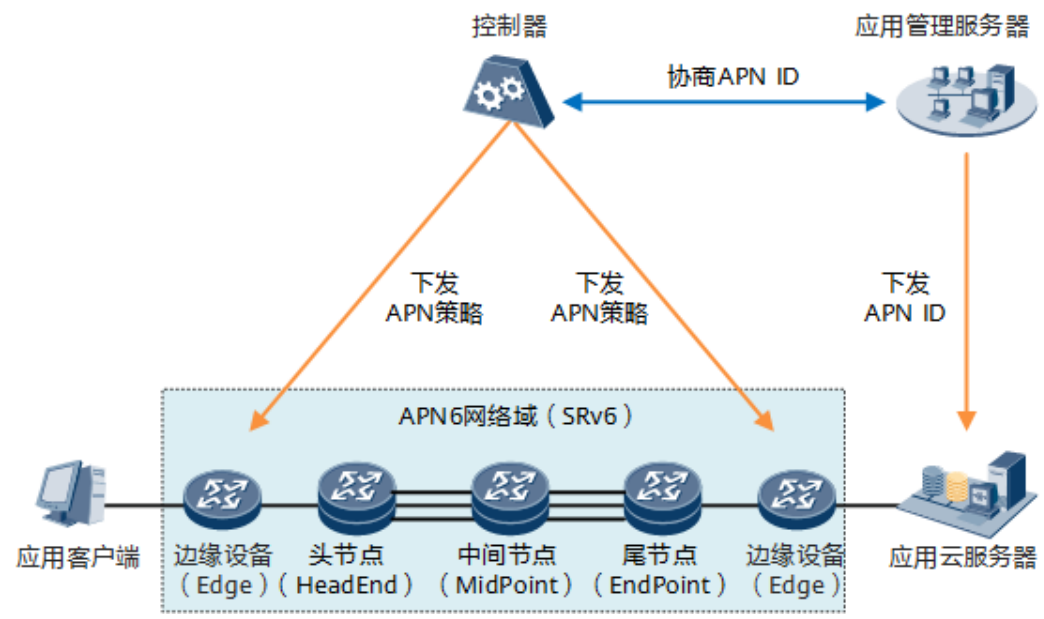


APN6 的网络架构

APN6的实现基础为IPv6，IPv6数据报文封装为APN6应用信息提供了可编程空间。如图1-3所示，APN6的网络架构中的组件包含：应用端侧/云侧设备、网络边缘设备、网络策略执行设备（头节点、中间节点、尾节点）、控制器。



图 1-3 APN6 的网络架构

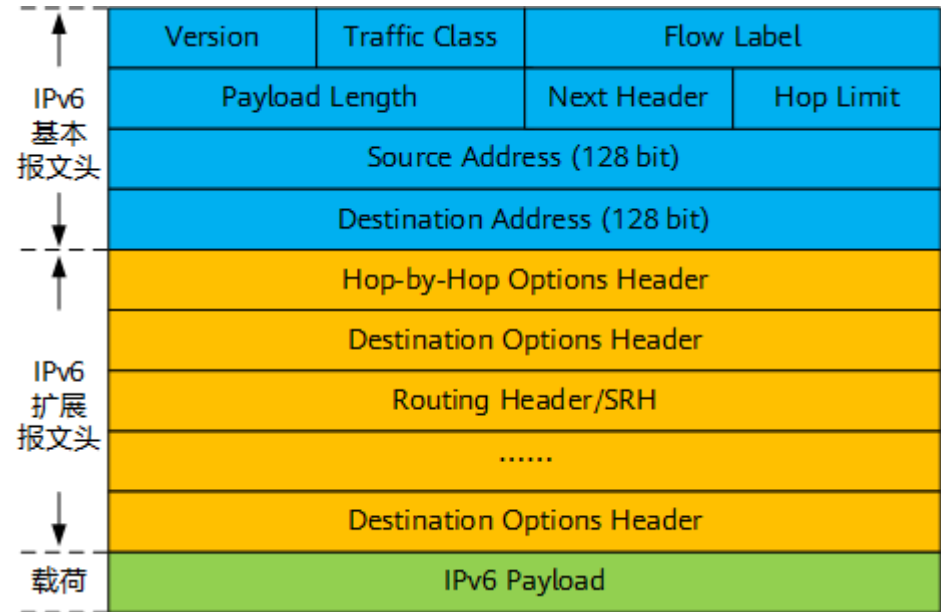


APN ID 由网络侧边缘设备生成，并封装在报文中，称为网络侧方案。APN ID 信息无需由应用端侧/云侧设备进行封装，而是由感知应用的网络边缘设备（如 CPE 等）根据预设策略进行封装，应用感知的信息来源为报文中的五元组信息和/或二层接口信息。在这种情况下，同样可以在 APN6 网络域内根据报文中所携带的应用信息提供相应的感知应用的精细网络服务。网络侧方案无需应用侧的生态支持，在网络运营商和行业网络中都容易部署。

APN6 的实现基础

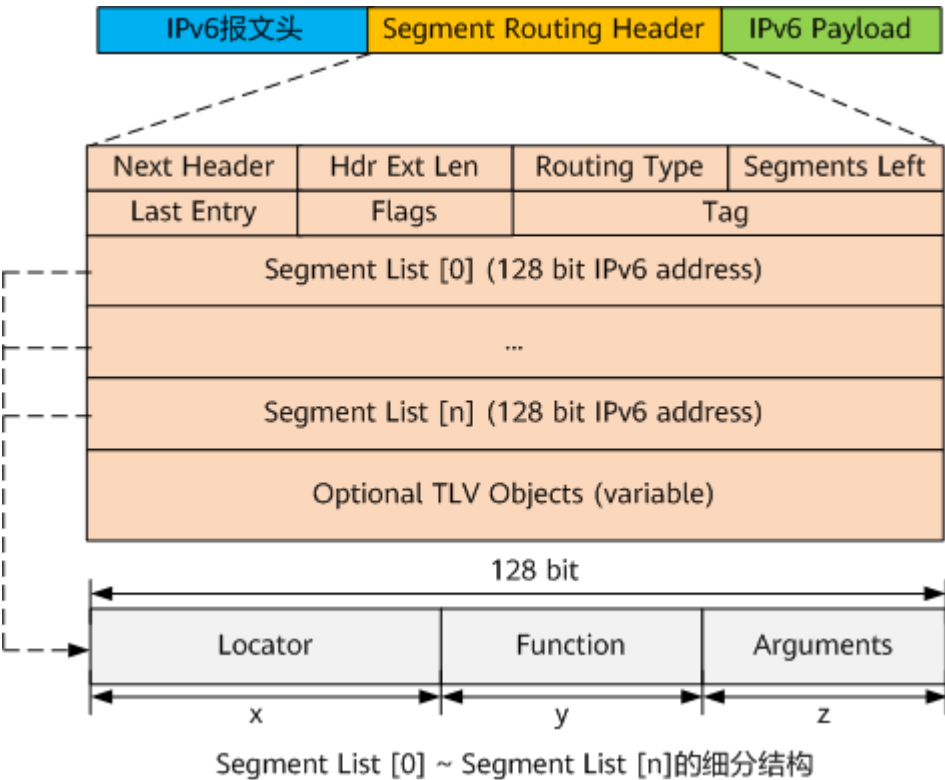
IPv6 扩展头的格式和推荐排序，如图 1-4 所示。其中，逐跳选项头 HBH（Hop-by-Hop Options Header）、目的选项头 DOH（Destination Options Header）、路由头 SRH（Segment Routing Header）提供可编程空间，可以用来携带应用信息。

图 1-4 IPv6 扩展头提供的可编程空间



如图1-5所示，SRv6 SRH的数据报文头具备多重可编程空间，可以用来携带应用信息。

图 1-5 SRH 提供的可编程空间



## APN6 的关键角色

### 端侧/云侧设备

通过应用感知的程序，感知应用的特征信息，生成封装了应用特征信息的数据报文，进入APN6网络域。

### 网络边缘设备

端侧/云侧设备不被要求必须具备应用感知能力，如果不具备，即无法发出封装了应用特征信息的数据报文。在这种情况下，将由网络边缘设备从五元组信息、业务信息（例如双VLAN标签C-VLAN和S-VLAN的映射）中解析出应用特征信息，并封装进数据报文，再转发至网络策略执行设备。

### 网络策略执行设备

在SRv6网络中，网络头节点和尾节点之间存在着一组能够满足不同SLA需求的网络服务路径。

- 感知应用的头节点：头节点负责维护入方向流量与网络服务路径的匹配关系。头节点从网络边缘设备接收到数据报文后，根据报文中携带的APN6应用信息，将流量匹配引入到满足需求的路径；也可以将应用信息复制并封装到外侧IPv6扩展头中，在SRv6网络中进一步提供感知应用服务。
- 感知应用的中间节点：中间节点根据头节点匹配的网络服务路径，为应用提供网络转发服务。同时，中间节点还可以根据报文中携带的APN6应用信息提供其他的网络增值服务，如感知应用的SFC、感知应用的IFIT等。

- 感知应用的尾节点：网络服务路径将在尾节点处终结，APN6应用信息可以在尾节点上和路径隧道封装被一起解除。在报文进入路径之前就已经存在的APN6应用信息，也可以随IPv6数据报文继续传输。

**控制器**

控制器可以统一规划和维护APN ID以及其与应用（组）、网络服务策略之间的映射关系，并且下发到网络边缘设备和网络策略执行设备，具体为：

- 对于网络边缘设备，控制器下发应用（组）和APN ID之间的映射关系。其中，应用可以基于原始报文的类型（分类规则如五元组、QinQ等）来进行分组。
- 对于网络策略执行设备，控制器下发报文中携带的APN ID和网络服务策略之间的映射关系。

**APN ID 生成**

IETF定义了APN报文所携带的应用信息（APN Attribute），包括应用标识信息（APN ID）和应用需求参数信息（APN Parameters）。

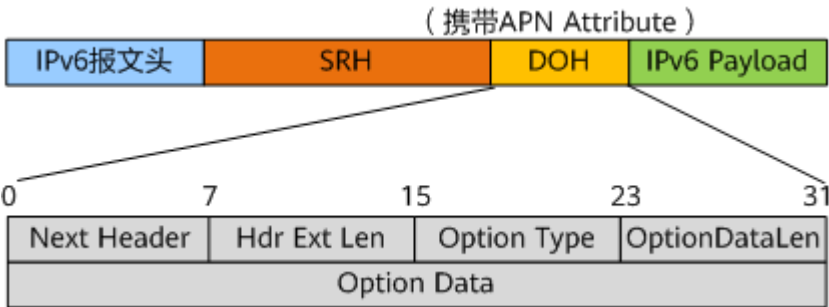
- APN ID：提供便于网络区分不同应用流和某个/类应用的不同用户（组）等信息，可以包括APP Group ID、User Group ID等信息。
- APN Parameters：可选携带应用关注的网络性能需求参数信息，包括带宽、时延、抖动、丢包率等，当前版本不支持该参数信息。

在网络侧方案中，应用信息可以由（行业）网络运营商，根据整体应用业务情况，进行统一规划。

**APN 标识位置**

APN6通过扩展IPv6数据平面，携带APN Attribute。APN Attribute携带在IPv6的DOH（Destination Options Header，目的选项报文头）中。DOH的报文头类型值为60。DOH通过Option Type，标识出APN6选项类型，来携带APN Attribute。可以用于携带APN Attribute的具体位置如图1-6所示。

图 1-6 APN 标识位置



Option区块字段解释如表1-1所示。

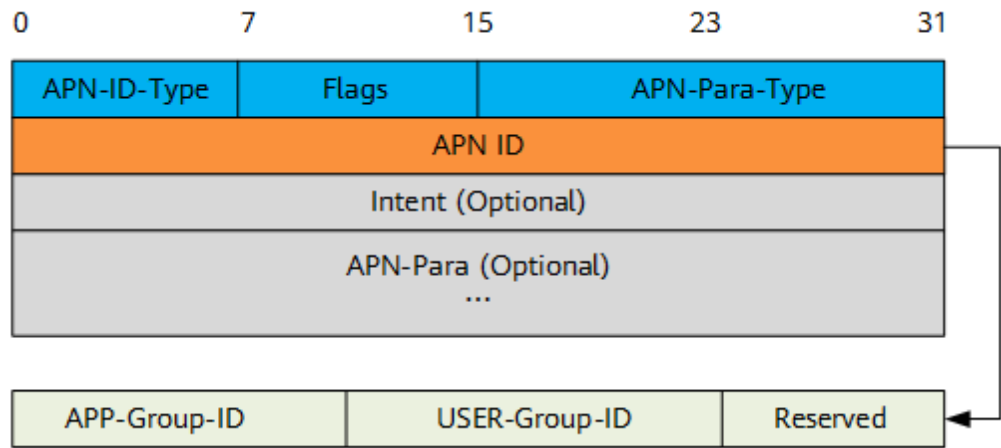
表 1-1 Option 区块字段解释

字段名	长度	含义
Option Type	8 bit	指明当前选项的类型，不同类型的选项，Data区域的数据格式不同。

字段名	长度	含义
Option Data Len	8 bit	指明当前选项Data区域的长度，单位为字节。
Option Data	长度可变	当前选项的数据内容。要使得整个目的选项报文头的长度为8字节的整数倍。数据长度不足时，可以使用填充选项。

IETF定义了APN Header的格式，如图1-7所示。

图 1-7 APN Header 的格式



各个字段的介绍如表1-2所示。

表 1-2 APN Header 各字段的含义

字段	含义
APN-ID-Type	1个字节，APN-ID的类型。
Flags	1个字节，当前未定义。
APN-Para-Type	2个字节，APN-Para的类型，描述APN-Para里包含了哪些网络性能的需求参数：带宽、时延、抖动、丢包率。暂未使用，填充保留值。
APN ID	应用唯一对应的的APN标识信息编号，由三部分组成： <ul style="list-style-type: none"><li>APP-Group-ID：应用组的标识信息，长度可变，由配置决定。</li><li>USER-Group-ID：用户组的标识信息，长度可变，由配置决定。</li><li>Reserved：预留字段。</li></ul>
Intent ( Optional )	可选，4个字节，描述向网络提出的一组意图需求。

字段	含义
APN-Para ( Optional )	可选，网络性能需求参数的具体内容信息，每个参数使用4个字节，包含哪些参数由APN-Para-Type决定。

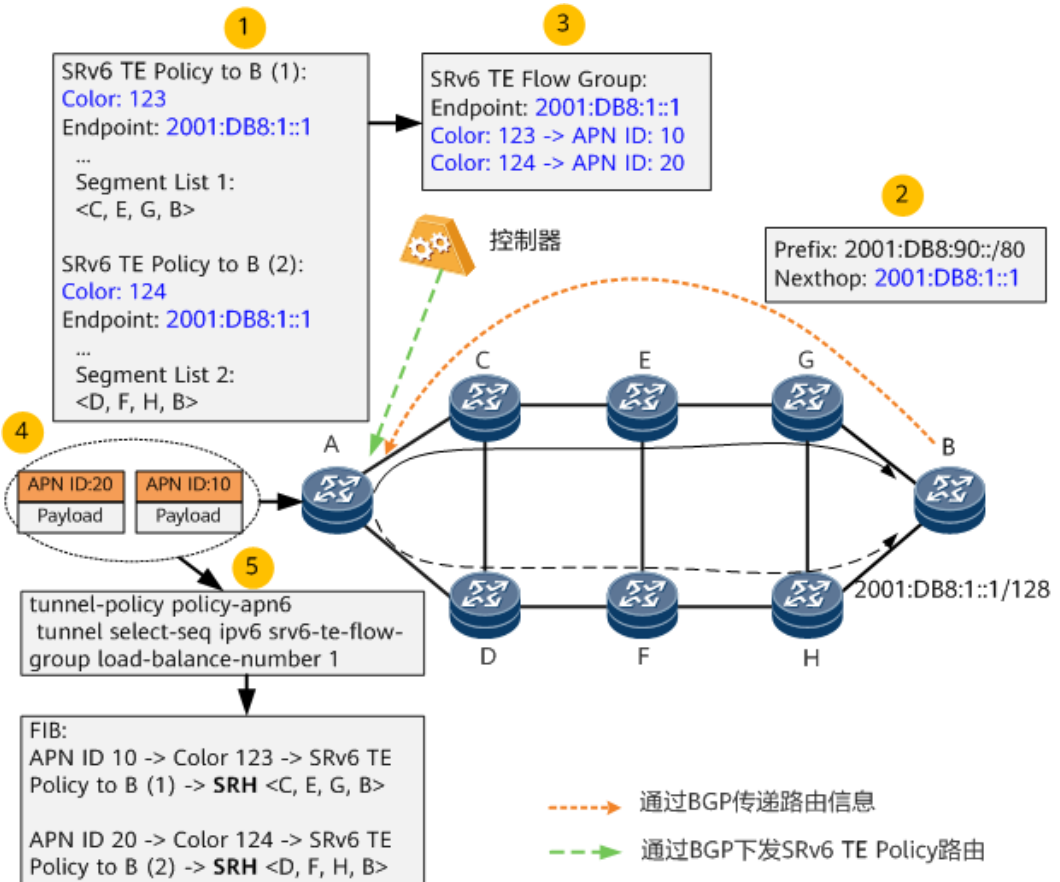
## APN ID 数据转发

### APN ID 引流

APN ID引流时，业务首先根据下一跳配置的隧道策略进行迭代，如果业务的迭代类型是SRv6 TE Flow Group，则头节点利用业务路由的Color属性匹配SRv6 Mapping Policy；如果SRv6 Mapping Policy存在，则头节点动态生成一个SRv6 TE Flow Group供业务转发使用。该SRv6 TE Flow Group里存在多个Color属性不同，但EndPoint相同的SRv6 TE Policy。如果根据业务的Color属性找不到对应的SRv6 Mapping Policy则不创建动态SRv6 TE Flow Group。如果有多个下一跳地址不同，但Color属性相同的业务期望迭代SRv6 TE Flow Group，则会根据一个SRv6 Mapping Policy的配置，创建出多个目的地址不同的动态SRv6 TE Flow Group。

下面以存在多color属性场景为例，描述APN ID引流的实现过程，如图1-8所示。

图 1-8 通过 APN ID 引流



引流过程简述如下：

1. 通过控制器向头端设备A下发SRv6 TE Policy，SRv6 TE Policy 1的Color是123，EndPoint是设备B的地址2001:DB8:1::1；SRv6 TE Policy 2的Color是124，EndPoint也是设备B的地址2001:DB8:1::1。
2. 设备B通过BGP邻居关系将BGP路由2001:DB8:90::/80发送给头端设备A。
3. 在头端设备A上进行隧道选择策略配置和SRv6 Mapping Policy配置，SRv6 Mapping Policy的映射类型为APN ID IPv6，同时定义了SRv6 TE Policy/Native IP和APN ID实例的映射规则。设备根据配置动态为业务生成SRv6 TE Flow Group。
4. 头端设备A上根据业务路由的下一跳地址已经关联到指定的SRv6 TE Flow Group。头端设备A感知应用、将APN ID封装在数据报文中。
5. 在转发时，数据报文携带APN ID信息，头端设备A根据SRv6 Mapping Policy配置，关联到指定的Color，进而关联到SRv6 TE Flow Group内具体的某一个SRv6 TE Policy，从而实现报文的APN ID引流。

在SRv6 Mapping Policy中，可以为流量指定SRv6 TE Policy或者Native IP与APN ID实例的映射规则，每个APN ID实例仅支持关联一个SRv6 TE Policy或者Native IP。只有处于Up状态的SRv6 TE Policy才能和APN ID实例形成映射规则。

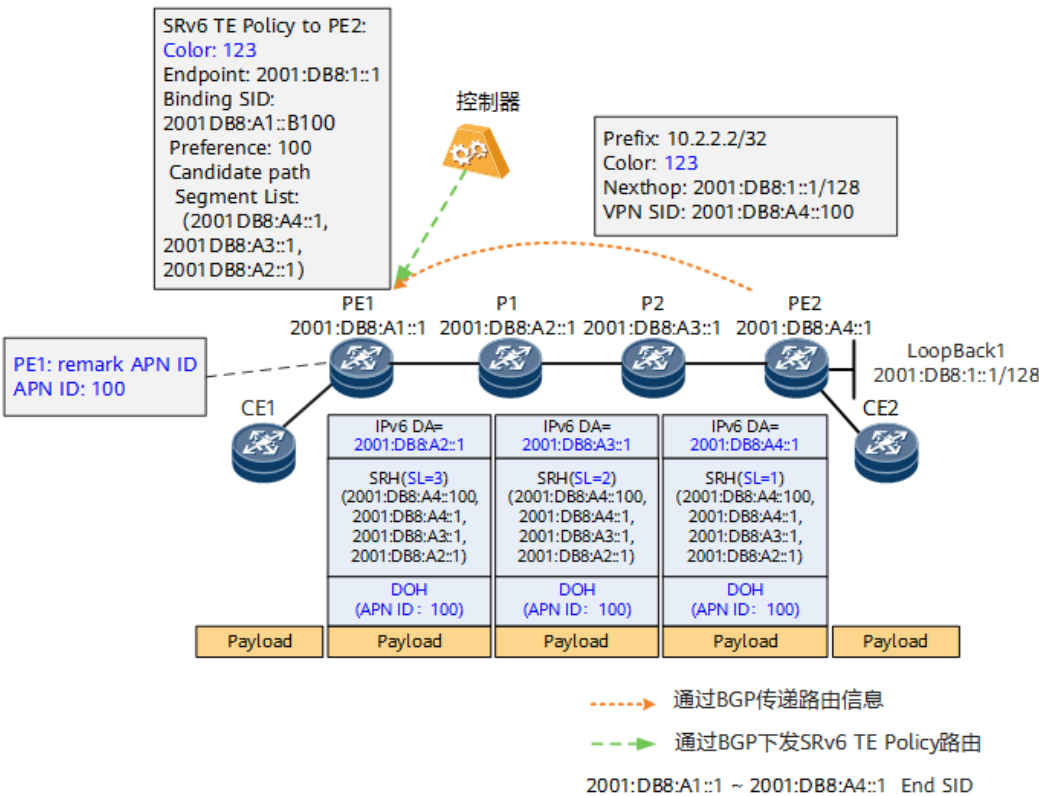
## APN6 数据转发

这里以L3VPNv4 over SRv6 TE Policy为例，介绍APN6在SRv6 TE Policy场景中的数据转发过程。具体如图1-9所示：

1. 控制器下发SRv6 TE Policy给头端PE1设备。
2. 尾端PE2设备发布BGP VPNv4路由10.2.2.2/32给头端PE1设备，BGP路由的下一跳是PE2设备的地址2001:DB8:1::1/128。
3. 在头端PE1设备上配置隧道策略。PE1设备在接收到BGP路由以后，利用路由的Color和下一跳迭代到SRv6 TE Policy，SRv6 TE Policy的SID List是<2001:DB8:A2::1, 2001:DB8:A3::1, 2001:DB8:A4::1>。其中SID List <2001:DB8:A2::1, 2001:DB8:A3::1, 2001:DB8:A4::1>用在数据转发场景时，也可以表示为(2001:DB8:A4::1, 2001:DB8:A3::1, 2001:DB8:A2::1)。
4. 在PE1设备上打上APN ID，值为100。
5. 头端PE1接收到CE1发送的普通单播报文后，查找VPN实例路由表，该路由的出接口是SRv6 TE Policy。PE1为报文插入SRH和DOH信息，封装SRv6 TE Policy的SID List，然后封装IPv6报文头信息。完成之后，PE1将报文对P1转发。
6. 中间P1和P2节点根据SRH信息逐跳转发。
7. 报文到达尾节点PE2之后，PE2使用报文的IPv6目的地址2001:DB8:A4::1查找My Local SID表，命中到End SID，所以PE2将报文SL减1，IPv6 DA更新为VPN SID 2001:DB8:A4::100。

PE2使用VPN SID 2001:DB8:A4::100查找My Local SID表，命中到End.DT4 SID，PE2解封装报文，去掉SRH信息、DOH信息和IPv6报文头，使用内层报文目的地址查找VPN SID 2001:DB8:A4::100对应的VPN实例路由表，然后将报文转发给CE2。

图 1-9 APN6 数据转发

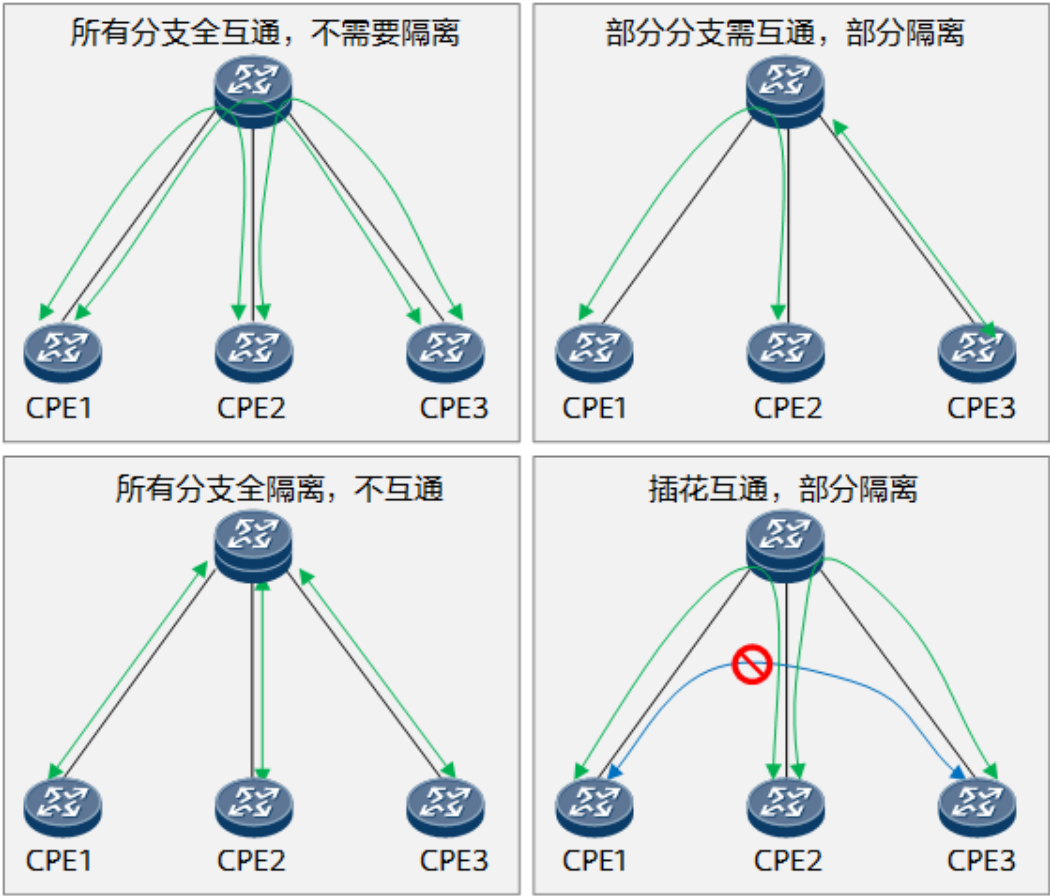


## APN ID 流量隔离

通过识别报文的APN ID，可以做业务的隔离控制。根据实际的隔离要求，隔离类型如图1-10所示。

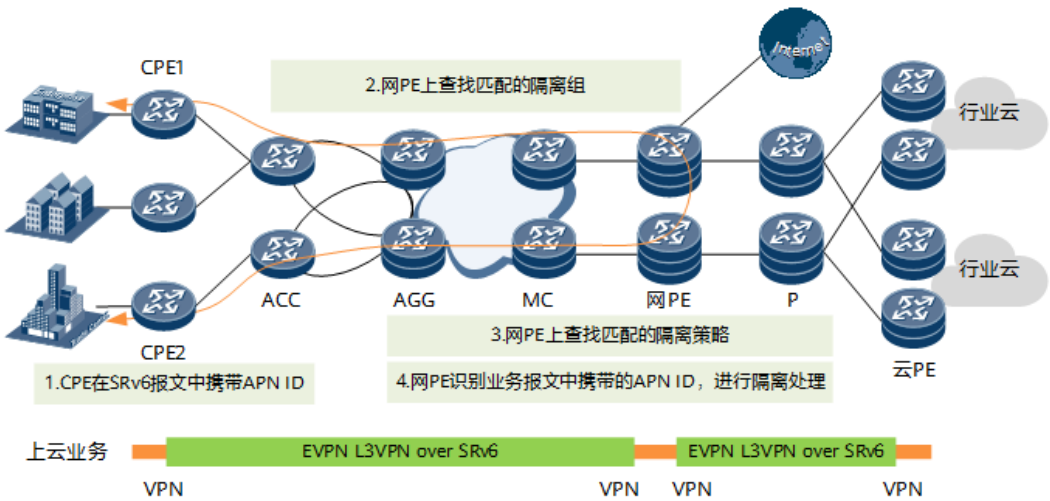


图 1-10 隔离类型



基于APN ID进行业务隔离的具体流程如图1-11所示。

图 1-11 隔离流程



1. CPE1配置本地APN ID封装策略，业务流量在CPE1上封装APN ID。
2. 网PE上配置VPN实例和对端Locator地址映射的隔离组（VPN出接口为公网）或接口下配置隔离组（VPN出接口为私网出接口），并在VPN实例下配置隔离策略。



流量到网PE后，基于VPN实例和对端Locator地址或出接口查找隔离组，再基于APN ID和隔离组匹配隔离策略；如果能匹配则进行流量隔离，否则正常转发。

### 1.1.1.1.3 APN6 应用

## APN6 在智能云网中的应用

由于企业侧上云从以前的单云模式向多云模式转型，企业用户已经使用部分云商的应用。运营商需要从以前单纯云接入向提供网+云的服务转型。运营商发挥网络差异化联接能力聚合多云，构筑云网的一体化接入、一体化服务和一体化运营，实现智联多云的目标。云骨干网络是实现这一目标的基础网络。运营商利用云骨干网络把公有云、私有云、第三方云、行业云等联接起来，通过云和网的物理预联接，实现云下一张网。

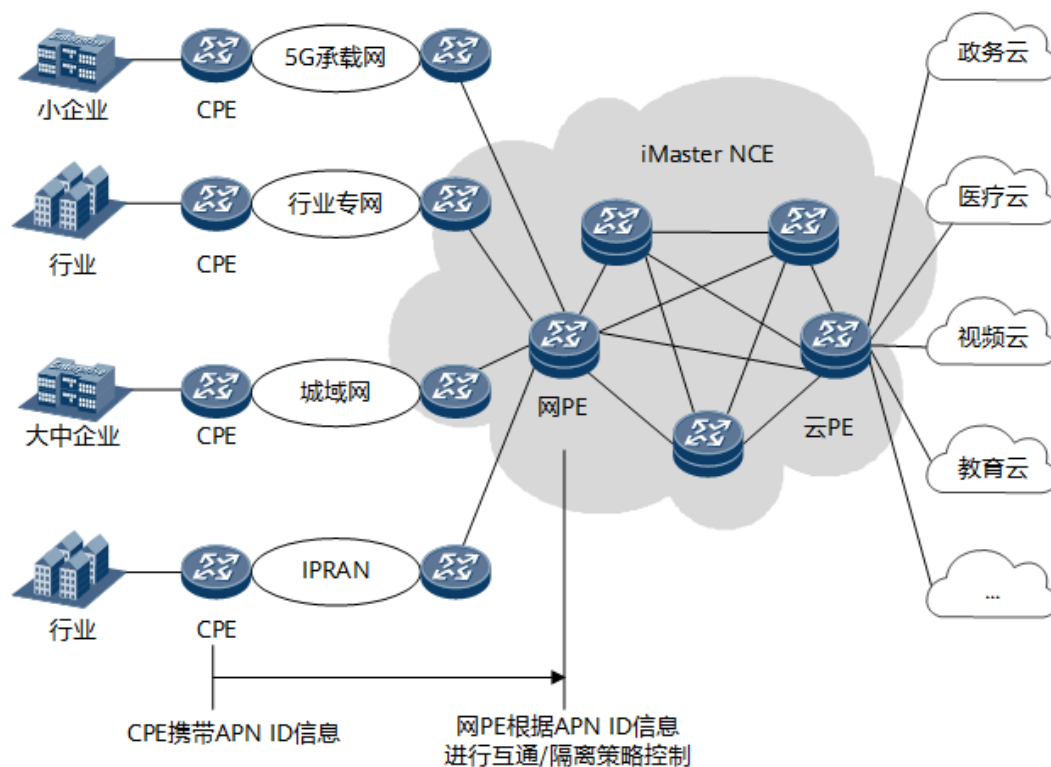
云骨干网络包括云PE和网PE两种重要类型的设备角色。

- 云PE负责联接云资源池，是云业务在运营商网络的联接点。
- 网PE负责联接运营商现有的云接入网络。

智能云网方案建议云PE和网PE之间Full Mesh组网，实现业务路径一跳直达。在实际项目应用中，云PE和网PE可能是一台物理设备，但是方案建议这两种设备角色独立设置。

智能云网方案如图1-12所示，通过APN6技术，可以支持MUX-VPN（Multiplex VPN，多路复用VPN）方案，满足全量接入场景（包括L3VPN、GRE、PW等）。

图 1-12 智能云网方案



从上图可以看出，CPE到网PE是一段隧道，网PE到云PE是一段隧道。CPE到网PE可支持多种接入场景：

1. 切片专网接入：EVPN over SRv6 TE Policy。
2. 城域接入：EVPN over SRv6 BE。
3. 5GtoB接入：GRE或SRv6 over GRE。
4. 存量IPRAN接入：PW或EVPN VPWS over SRv6接入。
5. 存量专线接入：Option A方案VLAN子接口接入。

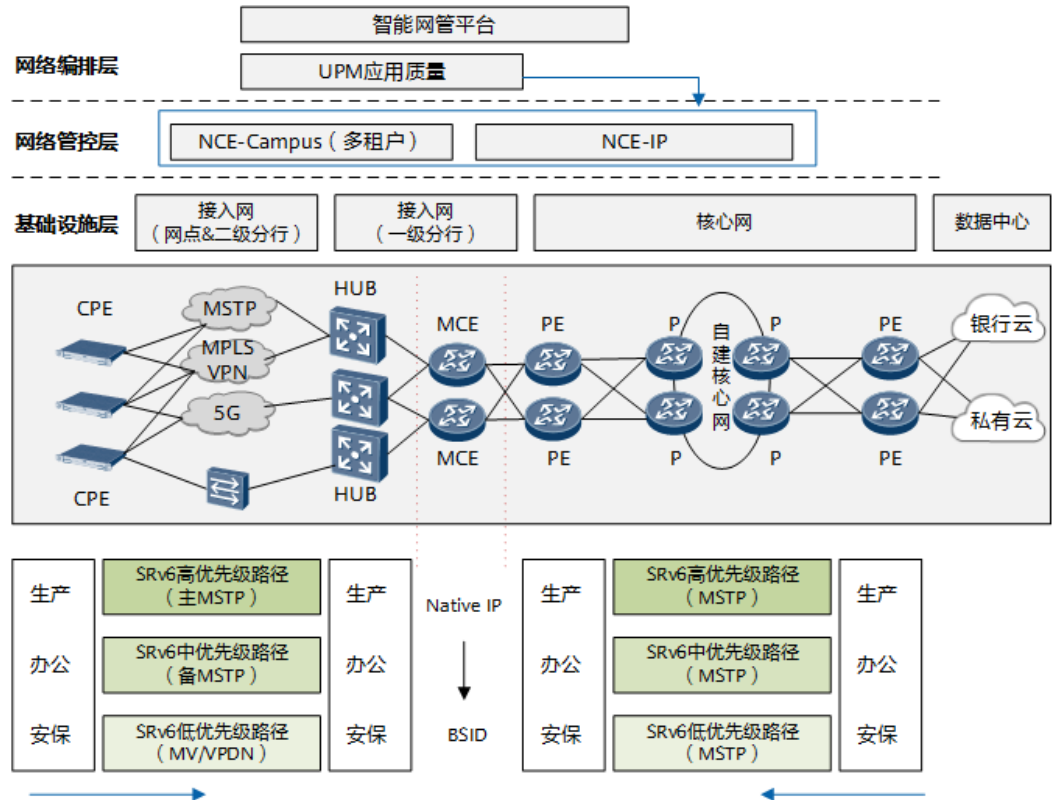
在该方案中，通过APN6可以实现策略控制CPE的隔离与互通，区分Separate和Group接口。Separate类型CPE相互隔离，Group类型CPE相互可互通。具体可实现的功能有：

1. CPE支持在SRv6报文中携带APN ID。
2. 网PE支持APN组策略配置。
3. 网PE支持识别业务报文中携带的APN ID，进行组策略处理。
4. 对于CPE到网PE不支持SRv6的场景，支持在网PE上基于GRE隧道口、PW的终结VE口等部署组ID。

## APN6 在金融行业中的应用

APN6在金融行业的应用如图1-13所示，主要体现在核心网部分，即核心网路由器设备打APN ID，识别APN ID。利用APN6技术，将应用信息（标识和网络性能需求）携带进入网络，使得网络感知应用及其需求，并为金融关键业务提供相应SLA保障，例如导入到SRv6 TE Policy路径等等。

图 1-13 金融行业应用



APN6在金融行业主要提供应用可视、特征画像、使能差异化服务、应用导流（入SRv6路径、切片、SFC）及应用调优等功能。APN6在金融行业的发展可以分为三个阶段：

- 解决应用可视问题。流量分类的方式为通过配置ACL规则，匹配IP地址、端口。这种方式存在两个问题：一是IP地址数量大时，ACL的规格不够，且配置工作量大；二是IP地址更新时，需要同步更新ACL的配置，存在ACL更新不及时的问题。
- 解决ACL规格过大无法维护的问题。随着业务规模增大，需要增加相应的ACL规则。日积月累，ACL规格逐步扩大，变得难以维护。通过在网络边缘为应用添加APN6标识，可以智能识别业务流，减小ACL规模。
- 端到端APN6，实现精准应用可视。跨域场景也可以跨租户模块拉通APN6配置。

### 1.1.1.2 APN6 配置

介绍了APN6的基本原理，基于APN6的配置过程和典型配置举例。

#### 1.1.1.2.1 APN6 概述

APN6（Application-aware IPv6 Networking，应用感知型IPv6网络）是一种新的网络架构，它利用IPv6报文自带的可编程空间，将应用信息（APN Attribute），包括应用标识信息（APN ID）和应用需求参数信息（APN Parameters）携带进入网络，进而为服务提供商提供精细的网络服务和精准的网络运维，当前版本应用信息中只支持携带APN ID。

APN6能够有效感知关键应用（组）、关键用户（组）、及其对网络的性能需求。对于服务提供商来说，SRv6、网络切片、DetNet、SFC、SD-WAN、IFIT等技术与APN6的结合，将极大地丰富云网服务维度，扩大云网商业增值空间，使能云网精细化运营。对于终端用户来说，可以享受更加精细的网络服务。

- 敏捷的自动选路：通过APN ID精细标识关键的应用（组）或用户（组），引导进入相应的SRv6 TE Policy，网络切片，或者SFC（Service Function Chain，业务功能链）路径等，实现应用分流和灵活选路。
- 应用流量可视监控：结合APN6，人工智能和大数据分析可以对关键应用或用户进行流量特征画像，呈现其流量路径、特征、变化规律及趋势，实现应用流量可视监控。
- 智能的网络运维：通过APN ID精细标识关键的应用（组）或用户（组），实施随路检测IFIT（逐跳、端到端两种选择模式），可以实现关键业务的实时性能监控。

#### 1.1.1.2.2 APN6 特性限制

#### 1.1.1.2.3 配置 APN6 基本功能

介绍了APN ID生成的基本配置。

### 配置 APN ID 模板

APN ID的模板配置主要用来定义一个APN ID的层次化结构模板。APN ID可以划分为app-group、user-group和保留字段三部分，APN ID的总长度、app-group和user-group的长度以及每个group下更细粒度的字段划分可以根据网络需求在模板中进行设计。模板配置完成后，APN ID实例可以引用配置的模板，为模板下的字段赋值，生成真正的APN ID。

## 前置任务

在配置APN6功能前，需要先申请APN6相关License，然后执行命令**license active**，激活带有APN6资源的License文件。

## 操作步骤

- 步骤1** 执行命令**system-view**，进入系统视图。
- 步骤2** 执行命令**apn**，创建并进入APN视图。
- 步骤3** 执行命令**ipv6**，创建并进入APN IPv6视图。
- 步骤4** （可选）执行命令**apn-id inherit { enable | disable }**，使能APN ID继承功能。
- 步骤5** 执行命令**apn-id template *template-name* [ length *total-length* { [ **app-group** *app-group-length* ] | [ **user-group** *user-group-length* ] } \*]**，创建并进入模板视图。
- 步骤6** 根据实际情况选择以下配置。
- 执行命令**app-group index *index-value* *field-name* length *field-length***，配置模板下app-group字段的名称和长度。
  - 执行命令**user-group index *index-value* *field-name* length *field-length***，配置模板下user-group字段的名称和长度。
- 步骤7** 执行命令**commit**，提交配置。
- 结束

## 配置 APN ID 实例

APN ID实例配置中主要是给APN ID模板各字段赋值。

## 操作步骤

- 步骤1** 执行命令**system-view**，进入系统视图。
- 步骤2** 执行命令**apn**，创建并进入APN视图。
- 步骤3** 执行命令**ipv6**，创建并进入APN IPv6视图。
- 步骤4** 执行命令**apn-id instance *instance-name***，创建并进入APN ID实例视图。
- 步骤5** 执行命令**template *template-name***，配置实例下引用的APN ID模板，该模板必须是在APN IPv6视图下已经存在的。
- 步骤6** 执行命令**apn-field *field-name* *field-value***，配置实例下app-group或者user-group对应的值。
- 步骤7** 执行命令**commit**，提交配置。
- 结束

## 配置流量策略

在应用感知引流场景中，设备根据报文目的地址查找FIB表获取Mapping Policy，然后根据Mapping Policy ID和APN标识确定报文的转发路径。为实现应用感知引流，需要先执行本章节配置流量策略。

## 操作步骤

**步骤1** 执行命令**system-view**，进入系统视图。

**步骤2** 定义流分类规则。

1. 根据ACL的类型，选择以下的命令创建ACL并进入ACL视图：

- 基本ACL，即acl-number的范围为2000～2999，请执行命令**acl { name basic-acl-name { basic | [ basic ] number basic-acl-number } | [ number ] basic-acl-number } [ match-order { config | auto } }**。
- 高级ACL，即acl-number的范围为3000～3999：请执行命令**acl { name advance-acl-name [ advance | [ advance ] number advance-acl-number ] | [ number ] advance-acl-number } [ match-order { config | auto } }**。

2. 根据ACL的类型，选择以下的命令创建ACL规则：

- 基本ACL，即acl-number的范围为2000～2999，请执行命令**rule [ rule-id ] [ name rule-name ] { deny | permit } [ fragment-type { fragment | non-fragment | non-subseq | fragment-subseq | fragment-spe-first } | source { source-ip-address { source-wildcard | 0 | src-netmask } | any } | time-range time-name | [ vpn-instance vpn-instance-name | vpn-instance-any ] | logging ] \***。
- 高级ACL，即acl-number的范围为3000～3999，根据不同的协议进行配置：  
**rule [ rule-id ] [ name rule-name ] { deny | permit } { protocol | tcp } [ [ dscp dscp | [ precedence precedence | tos tos ] \* ] | { destination { destination-ip-address { destination-wildcard | 0 | des-netmask } | any } | destination-pool destination-pool-name } | { destination-port operator port-number | destination-port-pool destination-port-pool-name } | fragment-type { fragment | non-fragment | non-subseq | fragment-subseq | fragment-spe-first } | { source { source-ip-address { source-wildcard | 0 | src-netmask } | any } | source-pool source-pool-name } | { source-port operator port-number | source-port-pool source-port-pool-name } | { tcp-flag | syn-flag } { tcp-flag [ mask mask-value ] | established } { ack [ fin | psh | rst | syn | urg ] \* } | { fin [ ack | psh | rst | syn | urg ] \* } | { psh [ fin | ack | rst | syn | urg ] \* } | { rst [ fin | psh | ack | syn | urg ] \* } | { syn [ fin | psh | rst | syn | urg ] \* } | { urg [ fin | psh | rst | syn | urg ] \* } } | time-range time-name | ttl ttl-operation ttl-value | packet-length length-operation length-value ] \***

此处以protocol为TCP协议为例进行介绍，其他协议的配置方法类似，详情可参考《命令参考》中的命令：**rule (高级ACL视图)(UDP)**、**rule (高级ACL视图)(ICMP)**、**rule (高级ACL视图)(gre-igmp-ip-ipv6-ospf)**。

### 说明

- 配置ACL规则时通常匹配源IP地址，如果要在ACL中添加多条规则，可以重复执行步骤2.2。
  - 同一ACL视图下的多条规则之间的匹配顺序可以配置为按深度优先（auto）或按配置优先（config）。缺省情况下，多条规则按照配置优先（config）的顺序进行匹配。  
通过ACL关联实例时，ACL规则中的地址通配符不支持非连续方式的子网掩码（子网掩码中的“0”和“1”位不连续，如255.0.255.0等），必须使用连续方式的子网掩码（子网掩码中的“0”和“1”位必须连续，如255.255.255.0等）。
3. 执行命令**commit**，提交配置。

4. 执行命令**quit**，回到系统视图。

### 步骤3 定义流分类。

1. 执行命令**traffic classifier** *classifier-name* [ **operator** { **and** | **or** } ]定义一个流分类并进入流分类视图。
2. 执行命令**if-match acl** *acl-number*，配置基于ACL列表进行复杂流分类的匹配规则。

如果要配置多条基于ACL列表的匹配规则，可以重复执行该步骤。

3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到系统视图。

### 步骤4 定义流行为。

1. 执行命令**traffic behavior** *behavior-name*，定义一个流行为并进入流行为视图。
2. 执行命令**remark apn-id-ipv6 instance** *apn-instance-name*，定义流量动作是为IPv6报文添加APN ID。
3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到系统视图。

### 步骤5 定义流策略。

1. 执行命令**traffic policy** *policy-name*，定义流量策略并进入策略视图。
2. 执行命令**classifier** *classifier-name* **behavior** *behavior-name*，在流量策略中为流分类指定采用的行为。
3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到系统视图。

### 步骤6 应用流分类策略。

#### 说明

匹配本地ACL策略生成APN ID后，只有路由迭代到SRv6 BE或者SRv6 TE Policy时支持把APN ID添加到报文中，其它隧道不支持添加APN ID。

在接口下应用流策略。

1. 执行命令**interface** *interface-type interface-number*，进入接口视图。
2. 执行命令**traffic-policy** *policy-name* **inbound** [ **link-layer** | **all-layer** | **mpls-layer** ]，在接口应用流量策略。
3. 执行命令**commit**，提交配置。

在VPN实例下应用流策略。

1. 执行命令**system-view**，进入系统视图。
2. 执行命令**ip vpn-instance** *vpn-instance-name*，进入VPN实例视图。
3. 执行命令**traffic-policy** *policy-name* **network inbound**，在VPN实例视图下应用流量策略。
4. 执行命令**commit**，提交配置。

----结束

## 检查配置结果

## 前提条件

配置APN6基本功能。

## 操作步骤

**步骤1** 在APN视图下，执行命令**display this**，查看APN6相关的配置信息。

**步骤2** 执行命令**display apn-id-ipv6 instance [ name *instance-name* ]**，查看APN ID实例的详细信息。

----结束

### 1.1.1.2.4（可选）配置 APN ID 引流

通过配置APN ID引流特性，可以使业务基于APN ID迭代到SRv6 TE Policy，利用SRv6 TE Policy里的路径信息指导数据转发。

## 前提条件

配置SRv6 TE Policy。

## 操作步骤

**步骤1** 执行命令**system-view**，进入系统视图。

**步骤2** 执行命令**segment-routing ipv6**，进入Segment Routing IPv6视图。

**步骤3** 执行命令**mapping-policy name-value color color-value**，创建SRv6 Mapping Policy，并且进入SRv6 Mapping Policy视图。

**步骤4** （可选）执行命令**description description-value**，配置SRv6 Mapping Policy的描述信息。

**步骤5** 执行命令**match-type apn-id-ipv6**，配置SRv6 Mapping Policy的映射类型为APN ID。

**步骤6** 根据实际情况选择如下命令，配置SRv6 TE Flow Group中SRv6 TE Policy或者Native IP和APN ID实例的映射。

- **index index-value instance instance-name match { srv6-te-policy color color-value | native-ip }**
- **default match { srv6-te-policy color color-value | native-ip }**

每个APN ID实例仅支持关联一个SRv6 TE Policy或者Native IP。只有处于Up状态的路径才能和APN ID实例形成映射。

携带APN ID值的流量入SRv6 TE Flow Group，按照如下规则进行转发：

1. 按照**index**命令的*index-value*进行顺序匹配，如果匹配成功，选择APN ID实例对应的SRv6 TE Policy或Native IP进行转发。

### 说明

如果多个APN ID实例对应的APN ID相同，按照*index-value*顺序选择第一个路径进行转发。



2. 若未匹配或对应的路径故障，则按照**default**命令对应的SRv6 TE Policy或Native IP进行转发。但若未配置**default**或者配置的**default**命令对应的路径不Up，流量会重新去选择**index-value**最小的且处于Up状态的路径进行转发。

#### 说明

建议低优先级的业务对应的APN ID实例配置在**index-value**小的映射规则中，保证APN ID未匹配的流量优先抢占低优先级业务的带宽。

**步骤7** 执行命令**quit**，退回Segment Routing IPv6视图。

**步骤8** 执行命令**quit**，退回系统视图。

**步骤9** 执行命令**tunnel-policy policy-name**，创建隧道策略，并进入Tunnel策略视图。

**步骤10** （可选）执行命令**description description-text**，对隧道策略配置描述信息。

**步骤11** 执行命令**tunnel select-seq ipv6 srv6-te-flow-group load-balance-number loadBalanceNumber**，配置隧道选择策略。

**步骤12** 执行命令**commit**，提交配置。

----结束

## 后续处理

1. 执行命令**display srv6-te flow-group**，查看SRv6 TE Flow Group的详细状态信息。
2. 执行命令**display srv6-te flow-group last-down-reason**，查看SRv6 TE Flow Group的down记录。

### 1.1.1.2.5 （可选）配置 APN ID 隔离策略

通过识别报文的APN ID信息，可以做业务的隔离控制。

## 前提条件

配置APN6基本功能。

## 操作步骤

**步骤1** 执行命令**system-view**，进入系统视图。

**步骤2** 执行命令**apn**，创建并进入APN视图。

**步骤3** 执行命令**ipv6**，创建并进入APN IPv6视图。

**步骤4** 执行命令**isolate-group name group-name**，配置APN隔离组。

**步骤5** （可选）配置VPN实例和隔离组之间的映射。

#### 说明

该步骤仅在出接口是SRv6隧道的场景下必须配置。

1. 执行命令**isolate-group mapping-vpn**，创建并进入隔离组VPN映射视图。
2. 执行命令**vpn-instance vpn-instance-name peer-locator peer-locator-value mask-length match isolate-group group-name**，配置VPN实例与隔离组之间的映射。



3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到APN IPv6视图。

**步骤6** 配置APN ID隔离策略。

1. 执行命令**apn-id isolate policy** *policy-name*，配置APN ID隔离策略并进入隔离策略视图。
2. （可选）执行命令**statistics enable**，配置APN ID隔离策略的统计功能。
3. 执行命令**index** *index-value* **instance** *instance-name* **isolate-group** *group-name* **behavior** *deny*，配置隔离策略内的APN ID实例与APN ID隔离组之间的行为规则。
4. 执行命令**commit**，提交配置。
5. 执行命令**quit**，回到APN IPv6视图。

**步骤7** 执行命令**quit**，回到APN视图。

**步骤8** 执行命令**quit**，回到系统视图。

**步骤9** 将相关接口加入到隔离组中。

1. 执行命令**interface** *interface-type interface-number*，进入接口视图。
2. 执行命令**apn-id-ipv6 isolate-group** *group-name*，在接口下指定对应的隔离组。
3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到系统视图。

**步骤10** 引用隔离策略。

1. 执行命令**ip vpn-instance** *vpn-instance-name*，进入VPN实例视图。
2. 执行命令**apn-id-ipv6 isolate-policy** *policy-name* **inbound**，在VPN实例下应用APN ID隔离策略。
3. 执行命令**commit**，提交配置。
4. 执行命令**quit**，回到系统视图。

----结束

## 后续处理

- 执行命令**display apn-id-ipv6 isolate-policy statistics** [ [ **policy** *policy-name* ] [ **vpn-instance** *instance-name* ] ]，查看VPN流量的隔离策略统计。

### 1.1.1.2.6 维护 APN6

在APN6网络网络中，可以通过Ping/Tracert来检查网络的连通性/故障发生的位置。

## 操作步骤

- 步骤1** 执行命令**ping ipv6** { **-apn-id-ipv6 instance** *instName* [ **-a** *source-ipv6-address* | **-c** *echo-number* | { **-s** *byte-number* | **-range** [ [ **min** *min-value* | **max** *max-value* | **step** *step-value* ] \* ] } | **-t** *timeout* | { **-tc** *traffic-class-value* | **-dscp** *dscp* } | **-vpn-instance** *vpn-instance-name* | **-m** *wait-time* | **-name** | **-si** { *source-interface-name* | *source-interface-type source-interface-number* } | { **-brief** | [ **-system-time** | **-ri** | **-detail** ] \* } | **-p** *pattern* ] \* *destination-ipv6-address* }，检测网络连接是否正常。

**步骤2** 执行命令 `tracert ipv6 -apn-id-ipv6 instance instName [ -f first-hop-limit | -m max-hop-limit | -p port-number | -q probes | -w timeout | vpn-instance vpn-instance-name | -s size | -name | -a source-ipv6-address | -v | -si { source-interface-name | source-interface-type source-interface-number } | { -tc tc | -dscp dscp } ] * host-name`, 测试故障发生的位置。

----结束

### 1.1.1.2.7 配置举例

#### 配置 APN6 在 L3VPNv4 over SRv6 TE Policy 场景的应用示例

介绍通过APN6在L3VPNv4 over SRv6 TE Policy场景应用的配置过程。

#### 组网需求

如图1-14所示:

- PE1、P1、P2和PE2属于同一自治系统，要求它们之间通过IS-IS协议达到IPv6网络互连的目的。
- PE1、P1、P2和PE2属于IS-IS进程1，都是Level-1设备。

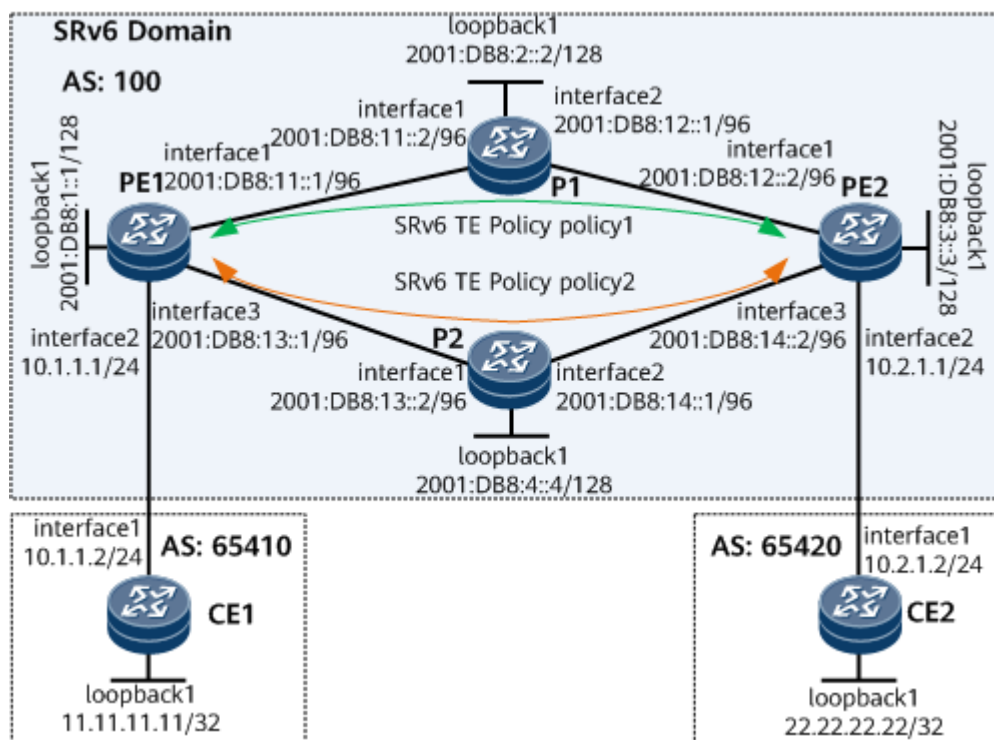
要求在PE1和PE2之间建立双向SRv6 TE Flow Group，承载L3VPNv4业务。

SRv6 TE Flow Group是实现APN ID引流的一种方式。在路由迭代时，头节点根据业务路由的下一跳地址关联到指定的SRv6 TE Flow Group。在数据转发时，业务携带APN ID信息，根据SRv6 Mapping Policy配置，关联到指定的Color，进而关联到SRv6 TE Flow Group内具体的某一个SRv6 TE Policy，从而实现报文的APN ID引流。

图 1-14 配置 L3VPNv4 over SRv6 TE Flow Group 组网图

#### 说明

本例中interface1、interface2、interface3分别代表GE1/0/0、GE2/0/0、GE3/0/0。



## 配置思路

采用如下的思路配置L3VPNv4 over SRv6 TE Flow Group示例：

1. 使能PE1、P1、P2和PE2各接口的IPv6转发能力，配置各接口的IPv6地址。
2. 在PE1、P1、P2和PE2上使能IS-IS，配置Level级别，指定网络实体。
3. 在PE1和PE2上配置VPN实例。
4. 在PE和CE之间建立EBGP对等体关系。
5. 在PE之间建立MP-IBGP对等体关系。
6. 在PE1、P1、P2和PE2上配置SRv6 SID，配置IS-IS的SRv6能力。同时在PE1和PE2上配置VPN私网路由携带SID属性。
7. 在PE1和PE2上配置SRv6 TE Policy。
8. 在PE1和PE2上配置APN6实例信息。
9. 在PE1和PE2上配置业务流的过滤策略，符合条件的业务流按照指定的方式生成APN ID信息。
10. 在PE1和PE2上配置SRv6 Mapping Policy。
11. 在PE1和PE2上配置隧道策略，优选SRv6 TE Flow Group，引入私网流量。

## 数据准备

为完成此配置例，需准备如下的数据：

- PE1、P1、P2和PE2各接口的IPv6地址。
- PE1、P1、P2和PE2的IS-IS进程号。
- PE1、P1、P2和PE2的IS-IS级别。
- PE1和PE2上VPN实例名称，VPN实例的RD和RT。

## 操作步骤

### 步骤1 使能各接口的IPv6转发能力，配置各接口的IPv6地址

# 配置PE1。P1、P2和PE2的配置过程与PE1类似，不再赘述，详情可参考配置文件。

```
<HUAWEI> system-view
[~HUAWEI] sysname PE1
[*HUAWEI] commit
[~PE1] interface gigabitethernet 1/0/0
[~PE1-GigabitEthernet1/0/0] ipv6 enable
[*PE1-GigabitEthernet1/0/0] ipv6 address 2001:DB8:11::1 96
[*PE1-GigabitEthernet1/0/0] quit
[*PE1] interface gigabitethernet 3/0/0
[*PE1-GigabitEthernet3/0/0] ipv6 enable
[*PE1-GigabitEthernet3/0/0] ipv6 address 2001:DB8:13::1 96
[*PE1-GigabitEthernet3/0/0] quit
[*PE1] interface LoopBack 1
[*PE1-LoopBack1] ipv6 enable
[*PE1-LoopBack1] ipv6 address 2001:DB8:1::1 128
[*PE1-LoopBack1] quit
[*PE1] commit
```

### 步骤2 配置IS-IS

# 配置PE1。

```
[~PE1] isis 1
[*PE1-isis-1] is-level level-1
[*PE1-isis-1] cost-style wide
[*PE1-isis-1] network-entity 10.0000.0000.0001.00
[*PE1-isis-1] ipv6 enable topology ipv6
[*PE1-isis-1] quit
[*PE1] interface gigabitethernet 1/0/0
[*PE1-GigabitEthernet1/0/0] isis ipv6 enable 1
[*PE1-GigabitEthernet1/0/0] quit
[*PE1] interface gigabitethernet 3/0/0
[*PE1-GigabitEthernet3/0/0] isis ipv6 enable 1
[*PE1-GigabitEthernet3/0/0] quit
[*PE1] interface loopback1
[*PE1-LoopBack1] isis ipv6 enable 1
[*PE1-LoopBack1] commit
[~PE1-LoopBack1] quit
```

# 配置P1。

```
[~P1] isis 1
[*P1-isis-1] is-level level-1
[*P1-isis-1] cost-style wide
[*P1-isis-1] network-entity 10.0000.0000.0002.00
[*P1-isis-1] ipv6 enable topology ipv6
[*P1-isis-1] quit
[*P1] interface gigabitethernet 1/0/0
[*P1-GigabitEthernet1/0/0] isis ipv6 enable 1
[*P1-GigabitEthernet1/0/0] quit
[*P1] interface gigabitethernet 2/0/0
[*P1-GigabitEthernet2/0/0] isis ipv6 enable 1
[*P1-GigabitEthernet2/0/0] quit
[*P1] interface loopback1
[*P1-LoopBack1] isis ipv6 enable 1
[*P1-LoopBack1] commit
[~P1-LoopBack1] quit
```

# 配置PE2。

```
[~PE2] isis 1
[*PE2-isis-1] is-level level-1
[*PE2-isis-1] cost-style wide
[*PE2-isis-1] network-entity 10.0000.0000.0003.00
```

```
[*PE2-isis-1] ipv6 enable topology ipv6
[*PE2-isis-1] quit
[*PE2] interface gigabitethernet 1/0/0
[*PE2-GigabitEthernet1/0/0] isis ipv6 enable 1
[*PE2-GigabitEthernet1/0/0] quit
[*PE2] interface gigabitethernet 3/0/0
[*PE2-GigabitEthernet3/0/0] isis ipv6 enable 1
[*PE2-GigabitEthernet3/0/0] quit
[*PE2] interface loopback1
[*PE2-LoopBack1] isis ipv6 enable 1
[*PE2-LoopBack1] commit
[~PE2-LoopBack1] quit
```

# 配置P2。

```
[~P2] isis 1
[*P2-isis-1] is-level level-1
[*P2-isis-1] cost-style wide
[*P2-isis-1] network-entity 10.0000.0000.0004.00
[*P2-isis-1] ipv6 enable topology ipv6
[*P2-isis-1] quit
[*P2] interface gigabitethernet 1/0/0
[*P2-GigabitEthernet1/0/0] isis ipv6 enable 1
[*P2-GigabitEthernet1/0/0] quit
[*P2] interface gigabitethernet 2/0/0
[*P2-GigabitEthernet2/0/0] isis ipv6 enable 1
[*P2-GigabitEthernet2/0/0] quit
[*P2] interface loopback1
[*P2-LoopBack1] isis ipv6 enable 1
[*P2-LoopBack1] commit
[~P2-LoopBack1] quit
```

配置完成后，可按如下指导检查IS-IS是否配置成功。

# 显示IS-IS邻居信息。以PE1为例。

```
[~PE1] display isis peer
```

```
Peer information for ISIS(1)

System Id      Interface      Circuit Id      State HoldTime Type  PRI
-----
0000.0000.0002* GE1/0/0      0000.0000.0003.01 Up 9s L1 64
0000.0000.0004* GE3/0/0      0000.0000.0004.02 Up 7s L1 64

Total Peer(s): 2
```

**步骤3** 在PE设备上配置使能IPv4地址族的VPN实例，将CE接入PE

# 配置PE1。

```
[~PE1] ip vpn-instance vpna
[*PE1-vpn-instance-vpna] ipv4-family
[*PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:1
[*PE1-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[*PE1-vpn-instance-vpna-af-ipv4] quit
[*PE1-vpn-instance-vpna] quit
[*PE1] interface gigabitethernet 2/0/0
[*PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[*PE1-GigabitEthernet2/0/0] ip address 10.1.1.1 24
[*PE1-GigabitEthernet2/0/0] quit
[*PE1] commit
```

# 配置PE2。

```
[~PE2] ip vpn-instance vpna
[*PE2-vpn-instance-vpna] ipv4-family
[*PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 200:1
[*PE2-vpn-instance-vpna-af-ipv4] vpn-target 111:1 both
[*PE2-vpn-instance-vpna-af-ipv4] quit
```

```
[*PE2-vpn-instance-vpna] quit
[*PE2] interface gigabitethernet 2/0/0
[*PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[*PE2-GigabitEthernet2/0/0] ip address 10.2.1.1 24
[*PE2-GigabitEthernet2/0/0] quit
[*PE2] commit
```

# 按图1-14配置各CE的接口IP地址，配置过程请参见后面的配置文件。

配置完成后，在PE设备上执行**display ip vpn-instance verbose**命令可以看到VPN实例的配置情况。各PE能ping通自己接入的CE。

#### 说明

当PE上有多个绑定了同一个VPN的接口，则使用**ping -vpn-instance**命令ping对端PE接入的CE时，要指定源IP地址，即要指定**ping -vpn-instance vpn-instance-name -a source-ip-address dest-ip-address**命令中的参数**-a source-ip-address**，否则可能ping不通。

### 步骤4 在PE与CE之间建立EBGP对等体关系

# 配置CE1。

```
[~CE1] interface loopback 1
[*CE1-LoopBack1] ip address 11.11.11.11 32
[*CE1-LoopBack1] quit
[*CE1] bgp 65410
[*CE1-bgp] peer 10.1.1.1 as-number 100
[*CE1-bgp] network 11.11.11.11 32
[*CE1-bgp] quit
[*CE1] commit
```

# 配置PE1。

```
[~PE1] bgp 100
[*PE1-bgp] router-id 1.1.1.1
[*PE1-bgp] ipv4-family vpn-instance vpna
[*PE1-bgp-vpna] peer 10.1.1.2 as-number 65410
[*PE1-bgp-vpna] import-route direct
[*PE1-bgp-vpna] commit
[~PE1-bgp-vpna] quit
[~PE1-bgp] quit
```

# 配置CE2。

```
[~CE2] interface loopback 1
[*CE2-LoopBack1] ip address 22.22.22.22 32
[*CE2-LoopBack1] quit
[*CE2] bgp 65420
[*CE2-bgp] peer 10.2.1.1 as-number 100
[*CE2-bgp] network 22.22.22.22 32
[*CE2-bgp] quit
[*CE2] commit
```

# 配置PE2。

```
[~PE2] bgp 100
[*PE2-bgp] router-id 2.2.2.2
[*PE2-bgp] ipv4-family vpn-instance vpna
[*PE2-bgp-vpna] peer 10.2.1.2 as-number 65420
[*PE2-bgp-vpna] import-route direct
[*PE2-bgp-vpna] commit
[~PE2-bgp-vpna] quit
[~PE2-bgp] quit
```

配置完成后，在PE设备上执行**display bgp vpnv4 vpn-instance peer**命令，可以看到PE与CE之间的BGP对等体关系已建立，并达到Established状态。

以PE1与CE1的对等体关系为例：

```
[~PE1] display bgp vpnv4 vpn-instance vpna peer
```

```
BGP local router ID : 1.1.1.1  
Local AS number : 100
```

```
VPN-Instance vpna, Router ID 1.1.1.1:  
Total number of peers : 1          Peers in established state : 1
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
10.1.1.2	4	65410	88	89	0	01:12:27	Established

## 步骤5 在PE之间建立MP-IBGP对等体关系

# 配置PE1。

```
[~PE1] bgp 100
```

```
[~PE1-bgp] peer 2001:DB8:3::3 as-number 100
```

```
[*PE1-bgp] peer 2001:DB8:3::3 connect-interface loopback 1
```

```
[*PE1-bgp] ipv4-family vpnv4
```

```
[*PE1-bgp-af-vpnv4] peer 2001:DB8:3::3 enable
```

```
[*PE1-bgp-af-vpnv4] commit
```

```
[~PE1-bgp-af-vpnv4] quit
```

```
[~PE1-bgp] quit
```

# 配置PE2。

```
[~PE2] bgp 100
```

```
[~PE2-bgp] peer 2001:DB8:1::1 as-number 100
```

```
[*PE2-bgp] peer 2001:DB8:1::1 connect-interface loopback 1
```

```
[*PE2-bgp] ipv4-family vpnv4
```

```
[*PE2-bgp-af-vpnv4] peer 2001:DB8:1::1 enable
```

```
[*PE2-bgp-af-vpnv4] commit
```

```
[~PE2-bgp-af-vpnv4] quit
```

```
[~PE2-bgp] quit
```

配置完成后，在PE设备上执行**display bgp vpnv4 all peer**命令，可以看到PE之间的BGP对等体关系已建立，并达到Established状态。

以PE1显示为例：

```
[~PE1] display bgp vpnv4 all peer
```

```
BGP local router ID : 1.1.1.1  
Local AS number : 100
```

```
Total number of peers : 2          Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
2001:DB8:3::3	4	100	85	85	0	01:10:30	Established

```
Peer of IPv4-family for vpn instance :
```

```
VPN-Instance vpna, Router ID 1.1.1.1:
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
10.1.1.2	4	65410	88	89	0	01:12:58	Established

## 步骤6 配置SRv6 SID，同时在PE上配置VPN私网路由携带SID属性

## 说明

End.DT4 SID可以通过BGP动态分配，也可以静态配置。静态配置和动态分配的SID同时存在时，静态配置优先生效。如果使用**segment-routing ipv6 locator locator-name**命令使能BGP动态分配End.DT4 SID或者使用**segment-routing ipv6 locator locator-name end-dt46**命令使能BGP动态分配End.DT46 SID，则不必执行命令**opcode func-opcode end-dt4 vpn-instance vpn-instance-name**或者**opcode func-opcode end-dt46 vpn-instance vpn-instance-name**配置静态SID的Opcode。

本例采用BGP动态分配方案。

### # 配置PE1。

```
[~PE1] segment-routing ipv6
[*PE1-segment-routing-ipv6] encapsulation source-address 2001:DB8:1::1
[*PE1-segment-routing-ipv6] locator as1 ipv6-prefix 2001:DB8:100:: 64 static 32
[*PE1-segment-routing-ipv6-locator] opcode ::100 end psp
[*PE1-segment-routing-ipv6-locator] opcode ::200 end no-flavor
[*PE1-segment-routing-ipv6-locator] quit
[*PE1-segment-routing-ipv6] quit
[*PE1] bgp 100
[*PE1-bgp] ipv4-family vpnv4
[*PE1-bgp-af-vpnv4] peer 2001:DB8:3::3 prefix-sid
[*PE1-bgp-af-vpnv4] quit
[*PE1-bgp] ipv4-family vpn-instance vpna
[*PE1-bgp-vpna] segment-routing ipv6 traffic-engineer best-effort
[*PE1-bgp-vpna] segment-routing ipv6 locator as1
[*PE1-bgp-vpna] commit
[~PE1-bgp-vpna] quit
[~PE1-bgp] quit
[~PE1] isis 1
[~PE1-isis-1] segment-routing ipv6 locator as1 auto-sid-disable
[*PE1-isis-1] commit
[~PE1-isis-1] quit
```

### # 配置P1。

```
[~P1] segment-routing ipv6
[*P1-segment-routing-ipv6] encapsulation source-address 2001:DB8:2::2
[*P1-segment-routing-ipv6] locator as1 ipv6-prefix 2001:DB8:200:: 64 static 32
[*P1-segment-routing-ipv6-locator] opcode ::100 end psp
[*P1-segment-routing-ipv6-locator] opcode ::200 end no-flavor
[*P1-segment-routing-ipv6-locator] quit
[*P1-segment-routing-ipv6] quit
[*P1] isis 1
[*P1-isis-1] segment-routing ipv6 locator as1 auto-sid-disable
[*P1-isis-1] commit
[~P1-isis-1] quit
```

### # 配置PE2。

```
[~PE2] segment-routing ipv6
[*PE2-segment-routing-ipv6] encapsulation source-address 2001:DB8:3::3
[*PE2-segment-routing-ipv6] locator as1 ipv6-prefix 2001:DB8:300:: 64 static 32
[*PE2-segment-routing-ipv6-locator] opcode ::100 end psp
[*PE2-segment-routing-ipv6-locator] opcode ::200 end no-flavor
[*PE2-segment-routing-ipv6-locator] quit
[*PE2-segment-routing-ipv6] quit
[*PE2] bgp 100
[*PE2-bgp] ipv4-family vpnv4
[*PE2-bgp-af-vpnv4] peer 2001:DB8:1::1 prefix-sid
[*PE2-bgp-af-vpnv4] quit
[*PE2-bgp] ipv4-family vpn-instance vpna
[*PE2-bgp-vpna] segment-routing ipv6 traffic-engineer best-effort
[*PE2-bgp-vpna] segment-routing ipv6 locator as1
[*PE2-bgp-vpna] commit
[~PE2-bgp-vpna] quit
[~PE2-bgp] quit
[~PE2] isis 1
```



```
[~PE2-isis-1] segment-routing ipv6 locator as1 auto-sid-disable
[*PE2-isis-1] commit
[~PE2-isis-1] quit
```

# 配置P2。

```
[~P2] segment-routing ipv6
[*P2-segment-routing-ipv6] encapsulation source-address 2001:DB8:4::4
[*P2-segment-routing-ipv6] locator as1 ipv6-prefix 2001:DB8:400:: 64 static 32
[*P2-segment-routing-ipv6-locator] opcode ::100 end psp
[*P2-segment-routing-ipv6-locator] opcode ::200 end no-flavor
[*P2-segment-routing-ipv6-locator] quit
[*P2-segment-routing-ipv6] quit
[*P2] isis 1
[*P2-isis-1] segment-routing ipv6 locator as1 auto-sid-disable
[*P2-isis-1] commit
[~P2-isis-1] quit
```

执行命令**display segment-routing ipv6 local-sid end forwarding**查看SRv6的Local SID表信息。

```
[~PE1] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----

SID      : 2001:DB8:100::100/128          FuncType  : End
Flavor   : PSP
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:24:18.785

SID      : 2001:DB8:100::200/128          FuncType  : End
Flavor   : NO-FLAVOR
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:24:18.785

Total SID(s): 2
[~P1] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----

SID      : 2001:DB8:200::100/128          FuncType  : End
Flavor   : PSP
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:29:22.633

SID      : 2001:DB8:200::200/128          FuncType  : End
Flavor   : NO-FLAVOR
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:29:22.633

Total SID(s): 2
[~PE2] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----

SID      : 2001:DB8:300::100/128          FuncType  : End
Flavor   : PSP
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:36:31.017

SID      : 2001:DB8:300::200/128          FuncType  : End
Flavor   : NO-FLAVOR
LocatorName : as1                      LocatorID  : 1
ProtocolType: STATIC                  ProcessID  : --
UpdateTime : 2021-05-05 09:36:31.017
```

```
Total SID(s): 2
[~P2] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----

SID      : 2001:DB8:400::100/128          FuncType  : End
Flavor   : PSP
LocatorName : as1                          LocatorID  : 1
ProtocolType: STATIC                      ProcessID  : --
UpdateTime : 2021-05-05 09:32:38.292

SID      : 2001:DB8:400::200/128          FuncType  : End
Flavor   : NO-FLAVOR
LocatorName : as1                          LocatorID  : 1
ProtocolType: STATIC                      ProcessID  : --
UpdateTime : 2021-05-05 09:32:38.292

Total SID(s): 2
```

## 步骤7 配置SRv6 TE Policy

# 配置PE1。

```
[~PE1] segment-routing ipv6
[~PE1-segment-routing-ipv6] segment-list list1
[*PE1-segment-routing-ipv6-segment-list-list1] index 5 sid ipv6 2001:DB8:200::100
[*PE1-segment-routing-ipv6-segment-list-list1] index 10 sid ipv6 2001:DB8:300::100
[*PE1-segment-routing-ipv6-segment-list-list1] quit
[*PE1-segment-routing-ipv6] segment-list list2
[*PE1-segment-routing-ipv6-segment-list-list2] index 5 sid ipv6 2001:DB8:400::100
[*PE1-segment-routing-ipv6-segment-list-list2] index 10 sid ipv6 2001:DB8:300::100
[*PE1-segment-routing-ipv6-segment-list-list2] quit
[*PE1-segment-routing-ipv6] srv6-te-policy locator as1
[*PE1-segment-routing-ipv6] srv6-te policy policy1 endpoint 2001:DB8:3::3 color 10
[*PE1-segment-routing-ipv6-policy-policy1] binding-sid 2001:DB8:100::900
[*PE1-segment-routing-ipv6-policy-policy1] candidate-path preference 100
[*PE1-segment-routing-ipv6-policy-policy1-path] segment-list list1
[*PE1-segment-routing-ipv6-policy-policy1-path] quit
[*PE1-segment-routing-ipv6-policy-policy1] quit
[*PE1-segment-routing-ipv6] srv6-te policy policy2 endpoint 2001:DB8:3::3 color 20
[*PE1-segment-routing-ipv6-policy-policy2] binding-sid 2001:DB8:100::901
[*PE1-segment-routing-ipv6-policy-policy2] candidate-path preference 100
[*PE1-segment-routing-ipv6-policy-policy2-path] segment-list list2
[*PE1-segment-routing-ipv6-policy-policy2-path] commit
[~PE1-segment-routing-ipv6-policy-policy2-path] quit
[~PE1-segment-routing-ipv6-policy-policy2] quit
[~PE1-segment-routing-ipv6] quit
```

# 配置PE2。

```
[~PE2] segment-routing ipv6
[~PE2-segment-routing-ipv6] segment-list list1
[*PE2-segment-routing-ipv6-segment-list-list1] index 5 sid ipv6 2001:DB8:200::100
[*PE2-segment-routing-ipv6-segment-list-list1] index 10 sid ipv6 2001:DB8:100::100
[*PE2-segment-routing-ipv6-segment-list-list1] quit
[*PE2-segment-routing-ipv6] segment-list list2
[*PE2-segment-routing-ipv6-segment-list-list2] index 5 sid ipv6 2001:DB8:400::100
[*PE2-segment-routing-ipv6-segment-list-list2] index 10 sid ipv6 2001:DB8:100::100
[*PE2-segment-routing-ipv6-segment-list-list2] quit
[*PE2-segment-routing-ipv6] srv6-te-policy locator as1
[*PE2-segment-routing-ipv6] srv6-te policy policy1 endpoint 2001:DB8:1::1 color 10
[*PE2-segment-routing-ipv6-policy-policy1] binding-sid 2001:DB8:300::900
[*PE2-segment-routing-ipv6-policy-policy1] candidate-path preference 100
[*PE2-segment-routing-ipv6-policy-policy1-path] segment-list list1
[*PE2-segment-routing-ipv6-policy-policy1-path] quit
[*PE2-segment-routing-ipv6-policy-policy1] quit
[*PE2-segment-routing-ipv6] srv6-te policy policy2 endpoint 2001:DB8:1::1 color 20
[*PE2-segment-routing-ipv6-policy-policy2] binding-sid 2001:DB8:300::901
[*PE2-segment-routing-ipv6-policy-policy2] candidate-path preference 100
```

```
[*PE2-segment-routing-ipv6-policy-policy2-path] segment-list list2
[*PE2-segment-routing-ipv6-policy-policy2-path] commit
[~PE2-segment-routing-ipv6-policy-policy2-path] quit
[~PE2-segment-routing-ipv6-policy-policy2] quit
[~PE2-segment-routing-ipv6] quit
```

配置完成后，执行命令**display srv6-te policy**查看SRv6 TE Policy信息。

以PE1的显示为例：

```
[~PE1] display srv6-te policy
PolicyName : policy1
Color      : 10                      Endpoint      : 2001:DB8:3::3
TunnelId   : 1
TunnelType  : SRv6-TE Policy          DelayTimerRemain : -
Policy State : Up                    State Change Time : 2021-05-05 09:24:24
Admin State : Up                    Traffic Statistics : Disable
Backup Hot-Standby : Disable          BFD           : Disable
Interface Index : -                  Interface Name   : -
Interface State : -                  Encapsulation Mode : Insert
Binding SID   : 2001:DB8:100::900(Insert, Preferred)
Candidate-path Count : 1

Candidate-path Preference : 100
Path State : Active          Path Type : Primary
Protocol-Origin : Configuration(30) Originator : 0, 0.0.0.0
Discriminator : 100          Binding SID : 2001:DB8:100::900
GroupID : 1                  Policy Name : policy1
Template ID : 0              Path Verification : Disable
DelayTimerRemain : -          Network Slice ID : -
Segment-List Count : 1
Segment-List : list1
Segment-List ID : 1          XcIndex : 1
List State : Up              DelayTimerRemain : -
Verification State : -        SuppressTimeRemain : -
PMTU : 9600                  Active PMTU : 9600
Weight : 1                    BFD State : -
Loop Detection State : Up      BFD Delay Remain : -
Network Slice ID : -
Binding SID : -
Reverse Binding ID : -
SID :
    2001:DB8:200::100
    2001:DB8:300::100

PolicyName : policy2
Color      : 20                      Endpoint      : 2001:DB8:3::3
TunnelId   : 2
TunnelType  : SRv6-TE Policy          DelayTimerRemain : -
Policy State : Up                    State Change Time : 2021-05-05 09:24:24
Admin State : Up                    Traffic Statistics : Disable
Backup Hot-Standby : Disable          BFD           : Disable
Interface Index : -                  Interface Name   : -
Interface State : -                  Encapsulation Mode : Insert
Binding SID   : 2001:DB8:100::901(Insert, Preferred)
Candidate-path Count : 1

Candidate-path Preference : 100
Path State : Active          Path Type : Primary
Protocol-Origin : Configuration(30) Originator : 0, 0.0.0.0
Discriminator : 100          Binding SID : 2001:DB8:100::901
GroupID : 2                  Policy Name : policy2
Template ID : 0              Path Verification : Disable
DelayTimerRemain : -          Network Slice ID : -
Segment-List Count : 1
Segment-List : list2
Segment-List ID : 2          XcIndex : 2
List State : Up              DelayTimerRemain : -
Verification State : -        SuppressTimeRemain : -
PMTU : 9600                  Active PMTU : 9600
```

```
Weight          : 1          BFD State          : -
Loop Detection State : Up      BFD Delay Remain   : -
Network Slice ID   : -
Binding SID        : -
Reverse Binding ID : -
SID :
    2001:DB8:400::100
    2001:DB8:300::100
```

## 步骤8 配置APN6实例

# 配置PE1。

```
[~PE1] apn
[*PE1-apn] ipv6
[*PE1-apn-ipv6] apn-id template tmplt1 length 64 app-group 16
[*PE1-apn-ipv6-template-tmplt1] app-group index 1 app-group1 length 16
[*PE1-apn-ipv6-template-tmplt1] quit
[*PE1-apn-ipv6] apn-id instance APN6-instance1
[*PE1-apn-ipv6-instance-APN6-instance1] template tmplt1
[*PE1-apn-ipv6-instance-APN6-instance1] apn-field app-group1 1
[*PE1-apn-ipv6-instance-APN6-instance1] commit
[~PE1-apn-ipv6-instance-APN6-instance1] quit
[~PE1-apn-ipv6] quit
[~PE1-apn] quit
```

# 配置PE2。

```
[~PE2] apn
[*PE2-apn] ipv6
[*PE2-apn-ipv6] apn-id template tmplt1 length 64 app-group 16
[*PE2-apn-ipv6-template-tmplt1] app-group index 1 app-group1 length 16
[*PE2-apn-ipv6-template-tmplt1] quit
[*PE2-apn-ipv6] apn-id instance APN6-instance1
[*PE2-apn-ipv6-instance-APN6-instance1] template tmplt1
[*PE2-apn-ipv6-instance-APN6-instance1] apn-field app-group1 1
[*PE2-apn-ipv6-instance-APN6-instance1] commit
[~PE2-apn-ipv6-instance-APN6-instance1] quit
[~PE2-apn-ipv6] quit
[~PE2-apn] quit
```

## 步骤9 配置业务流的过滤策略，符合条件的业务流按照指定的方式生成APN ID信息。

# 配置PE1。

```
[~PE1] acl number 3333
[*PE1-acl-advance-3333] rule 5 permit ip source 11.11.11.11 0 destination 22.22.22.22 0
[*PE1-acl-advance-3333] commit
[~PE1-acl-advance-3333] quit
[~PE1] traffic classifier c1
[*PE1-classifier-c1] if-match acl 3333
[*PE1-classifier-c1] commit
[~PE1-classifier-c1] quit
[~PE1] traffic behavior b1
[*PE1-behavior-b1] remark apn-id-ipv6 instance APN6-instance1
[*PE1-behavior-b1] commit
[~PE1-behavior-b1] quit
[~PE1] traffic policy p1
[*PE1-trafficpolicy-p1] classifier c1 behavior b1
[*PE1-trafficpolicy-p1] share-mode
[*PE1-trafficpolicy-p1] statistics enable
[*PE1-trafficpolicy-p1] quit
[*PE1] interface GigabitEthernet 2/0/0
[*PE1-GigabitEthernet2/0/0] traffic-policy p1 inbound
[*PE1-GigabitEthernet2/0/0] commit
[~PE1-GigabitEthernet2/0/0] quit
```

# 配置PE2。

```
[~PE2] acl number 3333
[*PE2-acl-advance-3333] rule 5 permit ip source 22.22.22.22 0 destination 11.11.11.11 0
```

```
[*PE2-acl-advance-3333] commit
[~PE2-acl-advance-3333] quit
[~PE2] traffic classifier c1
[*PE2-classifier-c1] if-match acl 3333
[*PE2-classifier-c1] commit
[~PE2-classifier-c1] quit
[~PE2] traffic behavior b1
[*PE2-behavior-b1] remark apn-id-ipv6 instance APN6-instance1
[*PE2-behavior-b1] commit
[~PE2-behavior-b1] quit
[~PE2] traffic policy p1
[*PE2-trafficpolicy-p1] classifier c1 behavior b1
[*PE2-trafficpolicy-p1] share-mode
[*PE2-trafficpolicy-p1] statistics enable
[*PE2-trafficpolicy-p1] quit
[*PE2] interface GigabitEthernet 2/0/0
[*PE2-GigabitEthernet2/0/0] traffic-policy p1 inbound
[*PE2-GigabitEthernet2/0/0] commit
[~PE2-GigabitEthernet2/0/0] quit
```

### 步骤10 配置SRv6 Mapping Policy

配置SRv6 Mapping Policy后，设备利用业务路由的Color属性去匹配相同Color的SRv6 Mapping Policy，如果SRv6 Mapping Policy存在，则设备动态生成一个SRv6 TE Flow Group，供业务转发使用。该SRv6 TE Flow Group里存在多个Color属性不同，但EndPoint相同的SRv6 TE Policy。

本例配置匹配APN6实例APN6-instance1的流量从policy1转发，其他流量默认从policy2转发。

# 配置PE1。

```
[~PE1] segment-routing ipv6
[~PE1-segment-routing-ipv6] mapping-policy p1 color 101
[*PE1-segment-routing-ipv6-mapping-policy-p1] match-type apn-id-ipv6
[*PE1-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] index 10 instance APN6-instance1 match
srv6-te-policy color 10
[*PE1-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] default match srv6-te-policy color 20
[*PE1-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] quit
[*PE1-segment-routing-ipv6-mapping-policy-p1] quit
[*PE1-segment-routing-ipv6] commit
[~PE1-segment-routing-ipv6] quit
```

# 配置PE2。

```
[~PE2] segment-routing ipv6
[~PE2-segment-routing-ipv6] mapping-policy p1 color 101
[*PE2-segment-routing-ipv6-mapping-policy-p1] match-type apn-id-ipv6
[*PE2-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] index 10 instance APN6-instance1 match
srv6-te-policy color 10
[*PE2-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] default match srv6-te-policy color 20
[*PE2-segment-routing-ipv6-mapping-policy-p1-apn-id-ipv6] quit
[*PE2-segment-routing-ipv6-mapping-policy-p1] quit
[*PE2-segment-routing-ipv6] commit
[~PE2-segment-routing-ipv6] quit
```

### 步骤11 配置隧道策略，引入私网流量

# 配置PE1。

```
[~PE1] route-policy p1 permit node 10
[*PE1-route-policy] apply extcommunity color 0:101
[*PE1-route-policy] quit
[*PE1] bgp 100
[*PE1-bgp] ipv4-family vpnv4
[*PE1-bgp-af-vpnv4] peer 2001:DB8:3::3 route-policy p1 import
[*PE1-bgp-af-vpnv4] quit
[*PE1-bgp] quit
```

```
[*PE1] tunnel-policy p1
[*PE1-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
[*PE1-tunnel-policy-p1] quit
[*PE1] ip vpn-instance vpna
[*PE1-vpn-instance-vpna] ipv4-family
[*PE1-vpn-instance-vpna-af-ipv4] tnl-policy p1
[*PE1-vpn-instance-vpna-af-ipv4] commit
[~PE1-vpn-instance-vpna-af-ipv4] quit
[~PE1-vpn-instance-vpna] quit
```

# 配置PE2。

```
[~PE2] route-policy p1 permit node 10
[*PE2-route-policy] apply extcommunity color 0:101
[*PE2-route-policy] quit
[*PE2] bgp 100
[*PE2-bgp] ipv4-family vpv4
[*PE2-bgp-af-vpv4] peer 2001:DB8:1::1 route-policy p1 import
[*PE2-bgp-af-vpv4] quit
[*PE2-bgp] quit
[*PE2] tunnel-policy p1
[*PE2-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
[*PE2-tunnel-policy-p1] quit
[*PE2] ip vpn-instance vpna
[*PE2-vpn-instance-vpna] ipv4-family
[*PE2-vpn-instance-vpna-af-ipv4] tnl-policy p1
[*PE2-vpn-instance-vpna-af-ipv4] commit
[~PE2-vpn-instance-vpna-af-ipv4] quit
[~PE2-vpn-instance-vpna] quit
```

配置完成后，执行**display srv6-te flow-group**命令，查看动态创建的SRv6 TE Flow Group状态。

以PE1的显示为例：

```
[~PE1] display srv6-te flow-group
```

```

SRv6-TE Flow Group Information
-----
Group Name      :
Color           : 101                Endpoint       : 2001:DB8:3::3
Group Tunnel ID : 3                  Group Tunnel Type : SRv6-TE Flow Group
Group Tunnel State : Up              State Change Time : 2021-05-05 09:38:19
Interface Index  : -                 Interface Name    : -
Interface State  : -
Delay Timer Remain : -                UP/ALL Num      : 2/2

Index           : -
APN ID IPv6 Instance : Default(Configure)
Match Tunnel     : SRv6-TE Policy      State          : Up
Color           : 20                  Tunnel Id       : 1

Index           : 1
APN ID IPv6 Instance : APN6-instance1
Match Tunnel     : SRv6-TE Policy      State          : Up
Color           : 10                  Tunnel Id       : 1
```

配置完成后，执行命令**display ip routing-table vpn-instance vpna**查看VPN实例路由表信息，可以看到私网路由已经成功迭代到SRv6 TE Policy。

以PE1的显示为例：

```
[~PE1] display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
-----
Routing Table : vpna
Destinations : 8    Routes : 8

Destination/Mask  Proto  Pre  Cost    Flags NextHop      Interface
```

```

10.1.1.0/24 Direct 0 0 D 10.1.1.1 GigabitEthernet2/0/0
10.1.1.1/32 Direct 0 0 D 127.0.0.1 GigabitEthernet2/0/0
10.1.1.255/32 Direct 0 0 D 127.0.0.1 GigabitEthernet2/0/0
10.2.1.0/24 IBGP 255 0 RD 2001:DB8:3::3 SRv6-TE Flow Group
11.11.11.11/32 EBGP 255 0 RD 10.1.1.2 GigabitEthernet2/0/0
22.22.22.22/32 IBGP 255 0 RD 2001:DB8:3::3 SRv6-TE Flow Group
127.0.0.0/8 Direct 0 0 D 127.0.0.1 InLoopBack0
255.255.255.255/32 Direct 0 0 D 127.0.0.1 InLoopBack0
[~PE1] display ip routing-table vpn-instance vpna 22.22.22.22 verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole
route
-----
Routing Table : vpna
Summary Count : 1

Destination: 22.22.22.22/32
Protocol: IBGP Process ID: 0
Preference: 255 Cost: 0
NextHop: 2001:DB8:3::3 Neighbour: 2001:DB8:3::3
State: Active Adv Relied Age: 01h04m44s
Tag: 0 Priority: low
Label: 3 QoSInfo: 0x0
IndirectID: 0x10000DE Instance:
RelayNextHop: :: Interface: SRv6-TE Flow Group
TunnelID: 0x000000003700002001 Flags: RD

```

## 步骤12 检查配置结果

同一VPN的CE能够相互Ping通，例如：

```

[~CE1] ping -a 11.11.11.11 22.22.22.22
PING 22.22.22.22: 56 data bytes, press CTRL_C to break
Reply from 22.22.22.22: bytes=56 Sequence=1 ttl=253 time=85 ms
Reply from 22.22.22.22: bytes=56 Sequence=2 ttl=253 time=33 ms
Reply from 22.22.22.22: bytes=56 Sequence=3 ttl=253 time=32 ms
Reply from 22.22.22.22: bytes=56 Sequence=4 ttl=253 time=30 ms
Reply from 22.22.22.22: bytes=56 Sequence=5 ttl=253 time=24 ms

--- 22.22.22.22 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 24/40/85 ms

```

----结束

## 配置文件

- PE1的配置文件

```

#
sysname PE1
#
ip vpn-instance vpna
ipv4-family
route-distinguisher 100:1
apply-label per-instance
tnl-policy p1
vpn-target 111:1 export-extcommunity
vpn-target 111:1 import-extcommunity
#
apn
ipv6
apn-id template tmplt1 length 64 app-group 16
app-group index 1 app-group1 length 16
apn-id instance APN6-instance1
template tmplt1
apn-field app-group1 1

```

```
#
acl number 3333
rule 5 permit ip source 11.11.11.11 0 destination 22.22.22.22 0
#
traffic classifier c1
if-match acl 3333
#
traffic behavior b1
remark apn-id-ipv6 instance APN6-instance1
#
traffic policy p1
share-mode
statistics enable
classifier c1 behavior b1 precedence 1
#
segment-routing ipv6
encapsulation source-address 2001:DB8:1::1
locator as1 ipv6-prefix 2001:DB8:100:: 64 static 32
opcode ::100 end psp
opcode ::200 end no-flavor
srv6-te-policy locator as1
segment-list list1
index 5 sid ipv6 2001:DB8:200::100
index 10 sid ipv6 2001:DB8:300::100
segment-list list2
index 5 sid ipv6 2001:DB8:400::100
index 10 sid ipv6 2001:DB8:300::100
srv6-te policy policy1 endpoint 2001:DB8:3::3 color 10
binding-sid 2001:DB8:100::900
candidate-path preference 100
segment-list list1
srv6-te policy policy2 endpoint 2001:DB8:3::3 color 20
binding-sid 2001:DB8:100::901
candidate-path preference 100
segment-list list2
mapping-policy p1 color 101
match-type apn-id-ipv6
index 10 instance APN6-instance1 match srv6-te-policy color 10
default match srv6-te-policy color 20
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0001.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator as1 auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:11::1/96
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ip binding vpn-instance vpna
ip address 10.1.1.1 255.255.255.0
traffic-policy p1 inbound
#
interface GigabitEthernet3/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:13::1/96
isis ipv6 enable 1
#
interface LoopBack1
ipv6 enable
```



```
ipv6 address 2001:DB8:1::1/128
isis ipv6 enable 1
#
bgp 100
router-id 1.1.1.1
peer 2001:DB8:3::3 as-number 100
peer 2001:DB8:3::3 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
#
ipv6-family unicast
undo synchronization
#
ipv4-family vpnv4
policy vpn-target
peer 2001:DB8:3::3 enable
peer 2001:DB8:3::3 route-policy p1 import
peer 2001:DB8:3::3 prefix-sid
#
ipv4-family vpn-instance vpna
import-route direct
segment-routing ipv6 locator as1
segment-routing ipv6 traffic-engineer best-effort
peer 10.1.1.2 as-number 65410
#
route-policy p1 permit node 10
apply extcommunity color 0:101
#
tunnel-policy p1
tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
#
return
```

- P1的配置文件

```
#
sysname P1
#
segment-routing ipv6
encapsulation source-address 2001:DB8:2::2
locator as1 ipv6-prefix 2001:DB8:200:: 64 static 32
opcode ::100 end psp
opcode ::200 end no-flavor
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0002.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator as1 auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:11::2/96
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:12::1/96
isis ipv6 enable 1
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:2::2/128
isis ipv6 enable 1
```

```
#
return
• PE2的配置文件
#
sysname PE2
#
ip vpn-instance vpna
  ipv4-family
    route-distinguisher 200:1
    apply-label per-instance
    tn1-policy p1
    vpn-target 111:1 export-extcommunity
    vpn-target 111:1 import-extcommunity
#
apn
  ipv6
    apn-id template tmplt1 length 64 app-group 16
    app-group index 1 app-group1 length 16
    apn-id instance APN6-instance1
    template tmplt1
    apn-field app-group1 1
#
acl number 3333
  rule 5 permit ip source 22.22.22.22 0 destination 11.11.11.11 0
#
traffic classifier c1
  if-match acl 3333
#
traffic behavior b1
  remark apn-id-ipv6 instance APN6-instance1
#
traffic policy p1
  share-mode
  statistics enable
  classifier c1 behavior b1 precedence 1
#
segment-routing ipv6
  encapsulation source-address 2001:DB8:3::3
  locator as1 ipv6-prefix 2001:DB8:300:: 64 static 32
  opcode ::100 end psp
  opcode ::200 end no-flavor
  srv6-te-policy locator as1
  segment-list list1
    index 5 sid ipv6 2001:DB8:200::100
    index 10 sid ipv6 2001:DB8:100::100
  segment-list list2
    index 5 sid ipv6 2001:DB8:400::100
    index 10 sid ipv6 2001:DB8:100::100
  srv6-te policy policy1 endpoint 2001:DB8:1::1 color 10
    binding-sid 2001:DB8:300::900
    candidate-path preference 100
    segment-list list1
  srv6-te policy policy2 endpoint 2001:DB8:1::1 color 20
    binding-sid 2001:DB8:300::901
    candidate-path preference 100
    segment-list list2
  mapping-policy p1 color 101
    match-type apn-id-ipv6
    index 10 instance APN6-instance1 match srv6-te-policy color 10
    default match srv6-te-policy color 20
#
isis 1
  is-level level-1
  cost-style wide
  network-entity 10.0000.0000.0003.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator as1 auto-sid-disable
#
```

```
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:12::2/96
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ip binding vpn-instance vpna
ip address 10.2.1.1 255.255.255.0
traffic-policy p1 inbound
#
interface GigabitEthernet3/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:14::2/96
isis ipv6 enable 1
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:3::3/128
isis ipv6 enable 1
#
bgp 100
router-id 2.2.2.2
peer 2001:DB8:1::1 as-number 100
peer 2001:DB8:1::1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
#
ipv6-family unicast
undo synchronization
#
ipv4-family vpnv4
policy vpn-target
peer 2001:DB8:1::1 enable
peer 2001:DB8:1::1 route-policy p1 import
peer 2001:DB8:1::1 prefix-sid
#
ipv4-family vpn-instance vpna
import-route direct
segment-routing ipv6 locator as1
segment-routing ipv6 traffic-engineer best-effort
peer 10.2.1.2 as-number 65420
#
route-policy p1 permit node 10
apply extcommunity color 0:101
#
tunnel-policy p1
tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
#
return
```

- P2的配置文件

```
#
sysname P2
#
segment-routing ipv6
encapsulation source-address 2001:DB8:4::4
locator as1 ipv6-prefix 2001:DB8:400:: 64 static 32
opcode ::100 end psp
opcode ::200 end no-flavor
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0004.00
#
ipv6 enable topology ipv6
```

```
segment-routing ipv6 locator as1 auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:13::2/96
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:14::1/96
isis ipv6 enable 1
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:4::4/128
isis ipv6 enable 1
#
return
```

- CE1的配置文件

```
#
sysname CE1
#
interface GigabitEthernet1/0/0
undo shutdown
ip address 10.1.1.2 255.255.255.0
#
interface LoopBack1
ip address 11.11.11.11 255.255.255.255
#
bgp 65410
peer 10.1.1.1 as-number 100
#
ipv4-family unicast
undo synchronization
network 11.11.11.11 255.255.255.255
peer 10.1.1.1 enable
#
return
```

- CE2的配置文件

```
#
sysname CE2
#
interface GigabitEthernet1/0/0
undo shutdown
ip address 10.2.1.2 255.255.255.0
#
interface LoopBack1
ip address 22.22.22.22 255.255.255.255
#
bgp 65420
peer 10.2.1.1 as-number 100
#
ipv4-family unicast
undo synchronization
network 22.22.22.22 255.255.255.255
peer 10.2.1.1 enable
#
return
```

## 配置 APN ID 隔离在 EVPN L3VPNv6 over SRv6 TE Policy 场景的应用示例

介绍APN ID隔离在EVPN L3VPNv6 over SRv6 TE Policy场景的配置过程。

## 组网需求

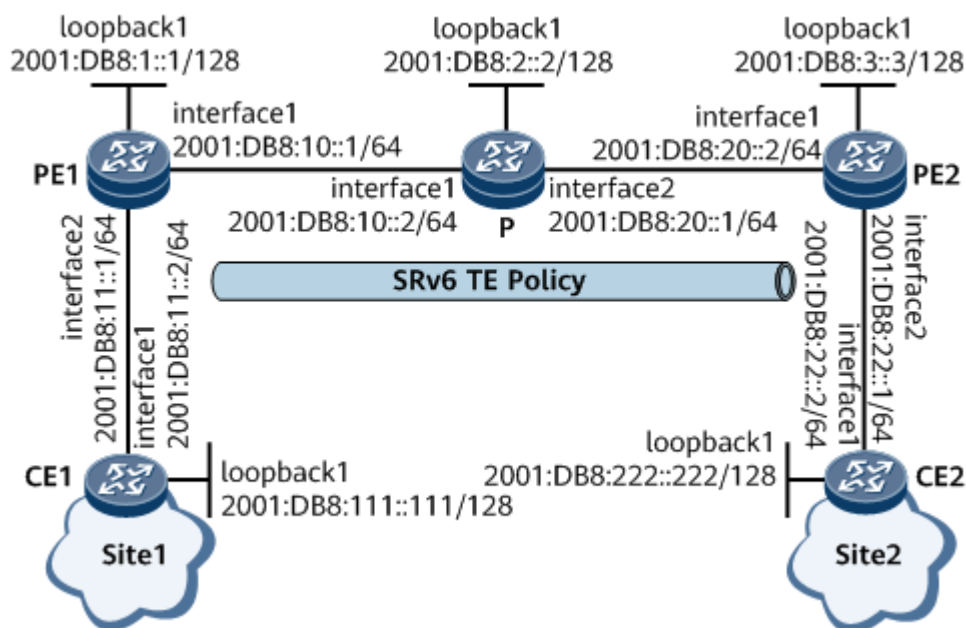
如图1-15所示：PE1、P和PE2属于同一自治系统，要求它们之间通过IS-IS协议达到IPv6网络互连的目的。在PE1和PE2之间通过建立双向SRv6 TE Policy承载EVPN L3VPNv6业务，在PE1上有三个流量入接口，通过两个接口进来的流量分别匹配不同的APN隔离策略进行流量隔离，只有一个接口进来的流量可以正常转发。

图 1-15 配置 EVPN L3VPNv6 over SRv6 TE Policy 组网图

### 说明

本例中interface1、interface2、interface3、interface4分别代表GE1/0/0、GE2/0/0、GE2/0/1、GE2/0/2。

interface3、interface4为PE设备和CE设备之间的互联口，和interface2类似，图中略。



## 配置注意事项

1. 配置SRv6 TE Policy需要使用End SID或End.X SID。SID可以通过手工静态配置，也可以由IGP动态生成。在静态配置SRv6 TE Policy场景中，如果使用动态SID，则SID在IGP协议重启后可能发生变化，此时静态SRv6 TE Policy也需要人工介入做相应的调整才能保持Up，在现网中实际无法大规模部署，基于上述原因，建议用户手工配置SID，禁止使用动态SID。
2. APN ID引流时，业务首先根据下一跳配置的隧道策略进行迭代，如果业务的迭代类型是SRv6 TE Flow Group，则头节点利用业务路由的Color属性匹配SRv6 Mapping Policy，如果SRv6 Mapping Policy存在，则头节点动态生成一个SRv6 TE Flow Group，供业务转发使用。该SRv6 TE Flow Group里存在多个Color属性不同，但EndPoint相同的SRv6 TE Policy。如果根据业务的Color属性找不到对应的SRv6 Mapping Policy时，则不创建动态SRv6 TE Flow Group。

## 配置思路

采用如下的思路配置EVPN L3VPNv6 over SRv6 TE Policy：

1. 使能PE1、P和PE2各接口的IPv6转发能力，配置各接口的IPv6地址。

2. 在PE1、P和PE2上使能IS-IS，配置Level级别，指定网络实体。
3. 在PE设备上配置EVPN L3VPN实例并将EVPN L3VPN实例绑定到接入侧接口。
4. 在PE和CE之间建立EBGP对等体关系。
5. 在PE设备之间建立BGP EVPN对等体关系。
6. 在PE1、P和PE2上配置SRv6 SID，配置IS-IS的SRv6能力。同时在PE1和PE2上配置VPN私网路由携带SID属性。
7. 在PE1和PE2上配置SRv6 TE Policy。
8. 在PE1和PE2上配置业务流的过滤策略，符合条件的业务流按照指定的方式生成APN ID信息。
9. 在PE1和PE2上配置并应用APN ID隔离策略。
10. 在PE1和PE2上配置业务流的APN ID信息。
11. 在PE1和PE2上配置隧道策略，引入私网流量。

## 数据准备

为完成此配置例，需准备如下的数据：

- PE1、P和PE2各接口的IPv6地址。
- PE1、P和PE2的IS-IS进程号。
- PE1、P和PE2的IS-IS级别。
- PE1和PE2上VPN实例名称，VPN实例的RD和RT。

## 操作步骤

### 步骤1 使能各接口的IPv6转发能力，配置各接口的IPv6地址

# 配置PE1。P和PE2的配置过程与PE1类似，不再赘述，详情可参考配置文件。

```
<HUAWEI> system-view
[~HUAWEI] sysname PE1
[*HUAWEI] commit
[~PE1] interface gigabitethernet 1/0/0
[~PE1-GigabitEthernet1/0/0] ipv6 enable
[*PE1-GigabitEthernet1/0/0] ipv6 address 2001:DB8:10::1 64
[*PE1-GigabitEthernet1/0/0] quit
[*PE1] interface LoopBack 1
[*PE1-LoopBack1] ipv6 enable
[*PE1-LoopBack1] ipv6 address 2001:DB8:1::1 128
[*PE1-LoopBack1] quit
[*PE1] commit
```

### 步骤2 配置IS-IS

# 配置PE1。

```
[~PE1] isis 1
[*PE1-isis-1] is-level level-1
[*PE1-isis-1] cost-style wide
[*PE1-isis-1] network-entity 10.0000.0000.0001.00
[*PE1-isis-1] ipv6 enable topology ipv6
[*PE1-isis-1] quit
[*PE1] interface gigabitethernet 1/0/0
[*PE1-GigabitEthernet1/0/0] isis ipv6 enable 1
[*PE1-GigabitEthernet1/0/0] quit
[*PE1] interface loopback1
[*PE1-LoopBack1] isis ipv6 enable 1
[*PE1-LoopBack1] quit
```

```
[*PE1] commit
```

# 配置P。

```
[~P] isis 1
[*P-isis-1] is-level level-1
[*P-isis-1] cost-style wide
[*P-isis-1] network-entity 10.0000.0000.0002.00
[*P-isis-1] ipv6 enable topology ipv6
[*P-isis-1] quit
[*P] interface gigabitethernet 1/0/0
[*P-GigabitEthernet1/0/0] isis ipv6 enable 1
[*P-GigabitEthernet1/0/0] quit
[*P] interface gigabitethernet 2/0/0
[*P-GigabitEthernet2/0/0] isis ipv6 enable 1
[*P-GigabitEthernet2/0/0] quit
[*P] interface loopback1
[*P-LoopBack1] isis ipv6 enable 1
[*P-LoopBack1] quit
[*P] commit
```

# 配置PE2。

```
[~PE2] isis 1
[*PE2-isis-1] is-level level-1
[*PE2-isis-1] cost-style wide
[*PE2-isis-1] network-entity 10.0000.0000.0003.00
[*PE2-isis-1] ipv6 enable topology ipv6
[*PE2-isis-1] quit
[*PE2] interface gigabitethernet 1/0/0
[*PE2-GigabitEthernet1/0/0] isis ipv6 enable 1
[*PE2-GigabitEthernet1/0/0] quit
[*PE2] interface loopback1
[*PE2-LoopBack1] isis ipv6 enable 1
[*PE2-LoopBack1] quit
[*PE2] commit
```

配置完成后，可按如下指导检查IS-IS是否配置成功。

# 显示IS-IS邻居信息。以PE1为例。

```
[~PE1] display isis peer
Peer information for ISIS(1)

System Id      Interface      Circuit Id      State HoldTime Type  PRI
-----
0000.0000.0002* GE1/0/0      0000.0000.0002.01 Up 8s L1 64

Total Peer(s): 1
```

### 步骤3 在PE设备上配置EVPN L3VPN实例并将EVPN L3VPN实例绑定到接入侧接口

# 配置PE1。

```
[~PE1] ip vpn-instance vpna
[*PE1-vpn-instance-vpna] ipv6-family
[*PE1-vpn-instance-vpna-af-ipv6] route-distinguisher 100:1
[*PE1-vpn-instance-vpna-af-ipv6] vpn-target 1:1 evpn
[*PE1-vpn-instance-vpna-af-ipv6] quit
[*PE1-vpn-instance-vpna] quit
[*PE1] interface gigabitethernet2/0/0
[*PE1-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[*PE1-GigabitEthernet2/0/0] ipv6 enable
[*PE1-GigabitEthernet2/0/0] ipv6 address 2001:DB8:11::1 64
[*PE1-GigabitEthernet2/0/0] quit
[*PE1] commit
```

# 配置PE2。

```
[~PE2] ip vpn-instance vpna
```

```
[*PE2-vpn-instance-vpna] ipv6-family
[*PE2-vpn-instance-vpna-af-ipv6] route-distinguisher 200:1
[*PE2-vpn-instance-vpna-af-ipv6] vpn-target 1:1 evpn
[*PE2-vpn-instance-vpna-af-ipv6] quit
[*PE2-vpn-instance-vpna] quit
[*PE2] interface gigabitethernet2/0/0
[*PE2-GigabitEthernet2/0/0] ip binding vpn-instance vpna
[*PE2-GigabitEthernet2/0/0] ipv6 enable
[*PE2-GigabitEthernet2/0/0] ipv6 address 2001:DB8:22::1 64
[*PE2-GigabitEthernet2/0/0] quit
[*PE2] commit
```

#### 步骤4 在PE与CE之间建立EBGP对等体关系

# 配置CE1。

```
[~CE1] interface loopback 1
[*CE1-LoopBack1] ipv6 enable
[*CE1-LoopBack1] ipv6 address 2001:DB8:111::111 128
[*CE1-LoopBack1] quit
[*CE1] bgp 65410
[*CE1-bgp] router-id 10.11.1.1
[*CE1-bgp] peer 2001:DB8:11::1 as-number 100
[*CE1-bgp] ipv6-family unicast
[*CE1-bgp-af-ipv6] peer 2001:DB8:11::1 enable
[*CE1-bgp-af-ipv6] import-route direct
[*CE1-bgp-af-ipv6] quit
[*CE1-bgp] quit
[*CE1] commit
```

# 配置PE1。

```
[~PE1] bgp 100
[~PE1-bgp] router-id 1.1.1.1
[*PE1-bgp] ipv6-family vpn-instance vpna
[*PE1-bgp-6-vpna] peer 2001:DB8:11::2 as-number 65410
[*PE1-bgp-6-vpna] import-route direct
[*PE1-bgp-6-vpna] advertise l2vpn evpn
[*PE1-bgp-6-vpna] commit
[~PE1-bgp-6-vpna] quit
[~PE1-bgp] quit
```

# 配置CE2。

```
[~CE2] interface loopback 1
[*CE2-LoopBack1] ipv6 enable
[*CE2-LoopBack1] ipv6 address 2001:DB8:222::222 128
[*CE2-LoopBack1] quit
[*CE2] bgp 65420
[*CE2-bgp] router-id 10.22.2.2
[*CE2-bgp] peer 2001:DB8:22::1 as-number 100
[*CE2-bgp] ipv6-family unicast
[*CE2-bgp-af-ipv6] peer 2001:DB8:22::1 enable
[*CE2-bgp-af-ipv6] import-route direct
[*CE2-bgp-af-ipv6] quit
[*CE2-bgp] quit
[*CE2] commit
```

# 配置PE2。

```
[~PE2] bgp 100
[~PE2-bgp] router-id 3.3.3.3
[*PE2-bgp] ipv6-family vpn-instance vpna
[*PE2-bgp-6-vpna] peer 2001:DB8:22::2 as-number 65420
[*PE2-bgp-6-vpna] import-route direct
[*PE2-bgp-6-vpna] advertise l2vpn evpn
[*PE2-bgp-6-vpna] commit
[~PE2-bgp-6-vpna] quit
[~PE2-bgp] quit
```



配置完成后，在PE设备上执行**display bgp vpnv6 vpn-instance peer**命令，可以看到PE与CE之间的BGP对等体关系已建立，并达到Established状态。

以PE1与CE1的对等体关系为例：

```
[~PE1] display bgp vpnv6 vpn-instance vpna peer
BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 1          Peers in established state : 1

VPN-Instance vpna, Router ID 1.1.1.1:
Peer      V      AS  MsgRcvd  MsgSent  OutQ  Up/Down    State  PrefRcv
2001:DB8:11::2  4    65410    131     132    0 01:51:37 Established    2
```

## 步骤5 在PE设备之间建立BGP EVPN对等体关系

# 配置PE1。

```
[~PE1] bgp 100
[~PE1-bgp] peer 2001:DB8:3::3 as-number 100
[*PE1-bgp] peer 2001:DB8:3::3 connect-interface loopback 1
[*PE1-bgp] l2vpn-family evpn
[*PE1-bgp-af-evpn] peer 2001:DB8:3::3 enable
[*PE1-bgp-af-evpn] quit
[*PE1-bgp] quit
[*PE1] commit
```

# 配置PE2。

```
[~PE2] bgp 100
[~PE2-bgp] peer 2001:DB8:1::1 as-number 100
[*PE2-bgp] peer 2001:DB8:1::1 connect-interface loopback 1
[*PE2-bgp] l2vpn-family evpn
[*PE2-bgp-af-evpn] peer 2001:DB8:1::1 enable
[*PE2-bgp-af-evpn] quit
[*PE2-bgp] quit
[*PE2] commit
```

配置完成后，在PE设备上执行**display bgp evpn peer**命令，可以看到PE之间的BGP EVPN对等体关系已建立，并达到Established状态。

以PE1的显示为例：

```
[~PE1] display bgp evpn peer

BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 1          Peers in established state : 1

Peer      V      AS  MsgRcvd  MsgSent  OutQ  Up/Down    State  PrefRcv
2001:DB8:3::3    4    100     40     40    0 00:30:41 Established    2
```

## 步骤6 配置SRv6 SID，同时在PE上配置VPN私网路由携带SID属性

# 配置PE1。

```
[~PE1] segment-routing ipv6
[*PE1-segment-routing-ipv6] encapsulation source-address 2001:DB8:1::1
[*PE1-segment-routing-ipv6] locator PE1 ipv6-prefix 2001:DB8:100:: 64 static 32
[*PE1-segment-routing-ipv6-locator] opcode ::10 end psp
[*PE1-segment-routing-ipv6-locator] quit
[*PE1-segment-routing-ipv6] quit
[*PE1] bgp 100
[*PE1-bgp] l2vpn-family evpn
[*PE1-bgp-af-evpn] peer 2001:DB8:3::3 advertise encap-type srv6
[*PE1-bgp-af-evpn] quit
[*PE1-bgp] ipv6-family vpn-instance vpna
[*PE1-bgp-6-vpna] segment-routing ipv6 traffic-engineer best-effort evpn
[*PE1-bgp-6-vpna] segment-routing ipv6 locator PE1 evpn
```

```
[*PE1-bgp-6-vpna] commit
[~PE1-bgp-6-vpna] quit
[~PE1-bgp] quit
[~PE1] isis 1
[~PE1-isis-1] segment-routing ipv6 locator PE1 auto-sid-disable
[*PE1-isis-1] commit
[~PE1-isis-1] quit
```

# 配置P。

```
[~P] segment-routing ipv6
[*P-segment-routing-ipv6] encapsulation source-address 2001:DB8:2::2
[*P-segment-routing-ipv6] locator P ipv6-prefix 2001:DB8:120:: 64 static 32
[*P-segment-routing-ipv6-locator] opcode ::20 end psp
[*P-segment-routing-ipv6-locator] quit
[*P-segment-routing-ipv6] quit
[*P] isis 1
[*P-isis-1] segment-routing ipv6 locator P auto-sid-disable
[*P-isis-1] commit
[~P-isis-1] quit
```

# 配置PE2。

```
[~PE2] segment-routing ipv6
[*PE2-segment-routing-ipv6] encapsulation source-address 2001:DB8:3::3
[*PE2-segment-routing-ipv6] locator PE2 ipv6-prefix 2001:DB8:130:: 64 static 32
[*PE2-segment-routing-ipv6-locator] opcode ::30 end psp
[*PE2-segment-routing-ipv6-locator] quit
[*PE2-segment-routing-ipv6] quit
[*PE2] bgp 100
[*PE2-bgp] l2vpn-family evpn
[*PE2-bgp-af-evpn] peer 2001:DB8:1::1 advertise encap-type srv6
[*PE2-bgp-af-evpn] quit
[*PE2-bgp] ipv6-family vpn-instance vpna
[*PE2-bgp-6-vpna] segment-routing ipv6 traffic-engineer best-effort evpn
[*PE2-bgp-6-vpna] segment-routing ipv6 locator PE2 evpn
[*PE2-bgp-6-vpna] commit
[~PE2-bgp-6-vpna] quit
[~PE2-bgp] quit
[~PE2] isis 1
[~PE2-isis-1] segment-routing ipv6 locator PE2 auto-sid-disable
[*PE2-isis-1] commit
[~PE2-isis-1] quit
```

执行命令**display segment-routing ipv6 local-sid end forwarding**查看SRv6的Local SID表信息。

```
[~PE1] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----
SID      : 2001:DB8:100::10/128          FuncType : End
Flavor   : PSP
LocatorName : PE1                      LocatorID: 1
ProtocolType: STATIC                   ProcessID: --
UpdateTime : 2021-08-30 01:46:05.713

Total SID(s): 1
[~PE2] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----
SID      : 2001:DB8:130::30/128          FuncType : End
Flavor   : PSP
LocatorName : PE2                      LocatorID: 1
ProtocolType: STATIC                   ProcessID: --
UpdateTime : 2021-08-30 01:47:26.426

Total SID(s): 1
```

```
[~P] display segment-routing ipv6 local-sid end forwarding
My Local-SID End Forwarding Table
-----
```

```
SID      : 2001:DB8:120::20/128          FuncType : End
Flavor   : PSP
LocatorName : P                          LocatorID: 1
ProtocolType: STATIC                     ProcessID: --
UpdateTime : 2021-08-30 01:49:44.292

Total SID(s): 1
```

## 步骤7 配置SRv6 TE Policy

# 配置PE1。

```
[~PE1] segment-routing ipv6
[~PE1-segment-routing-ipv6] segment-list list1
[*PE1-segment-routing-ipv6-segment-list-list1] index 5 sid ipv6 2001:DB8:120::20
[*PE1-segment-routing-ipv6-segment-list-list1] index 10 sid ipv6 2001:DB8:130::30
[*PE1-segment-routing-ipv6-segment-list-list1] commit
[~PE1-segment-routing-ipv6-segment-list-list1] quit
[~PE1-segment-routing-ipv6] srv6-te-policy locator PE1
[*PE1-segment-routing-ipv6] srv6-te policy policy1 endpoint 2001:DB8:3::3 color 101
[*PE1-segment-routing-ipv6-policy-policy1] binding-sid 2001:DB8:100::450
[*PE1-segment-routing-ipv6-policy-policy1] candidate-path preference 100
[*PE1-segment-routing-ipv6-policy-policy1-path] segment-list list1
[*PE1-segment-routing-ipv6-policy-policy1-path] commit
[~PE1-segment-routing-ipv6-policy-policy1-path] quit
[~PE1-segment-routing-ipv6-policy-policy1] quit
[~PE1-segment-routing-ipv6] quit
```

# 配置PE2。

```
[~PE2] segment-routing ipv6
[~PE2-segment-routing-ipv6] segment-list list1
[*PE2-segment-routing-ipv6-segment-list-list1] index 5 sid ipv6 2001:DB8:120::20
[*PE2-segment-routing-ipv6-segment-list-list1] index 10 sid ipv6 2001:DB8:100::10
[*PE2-segment-routing-ipv6-segment-list-list1] commit
[~PE2-segment-routing-ipv6-segment-list-list1] quit
[~PE2-segment-routing-ipv6] srv6-te-policy locator PE2
[*PE2-segment-routing-ipv6] srv6-te policy policy1 endpoint 2001:DB8:1::1 color 101
[*PE2-segment-routing-ipv6-policy-policy1] binding-sid 2001:DB8:130::350
[*PE2-segment-routing-ipv6-policy-policy1] candidate-path preference 100
[*PE2-segment-routing-ipv6-policy-policy1-path] segment-list list1
[*PE2-segment-routing-ipv6-policy-policy1-path] commit
[~PE2-segment-routing-ipv6-policy-policy1-path] quit
[~PE2-segment-routing-ipv6-policy-policy1] quit
[~PE2-segment-routing-ipv6] quit
```

配置完成后，执行命令**display srv6-te policy**查看SRv6 TE Policy信息。

以PE1的显示为例：

```
[~PE1] display srv6-te policy
PolicyName : policy1
Color      : 101                      Endpoint      : 2001:DB8:3::3
TunnelId   : 1
TunnelType : SRv6-TE Policy           DelayTimerRemain : -
Policy State : Up                     State Change Time : 2019-03-24 19:45:30
Admin State : Up                     Traffic Statistics : Disable
Backup Hot-Standby : Disable          BFD           : Disable
Interface Index : -                   Interface Name   : -
Interface State : -                   Encapsulation Mode : Insert
Binding SID   : 2001:DB8:100::450(Insert, Preferred)
Candidate-path Count : 1

Candidate-path Preference : 100
Path State                : Active          Path Type      : Primary
Protocol-Origin           : Configuration(30) Originator    : 0, 0.0.0.0
```

```

Discriminator      : 100
GroupID            : 1
Template ID        : 0
DelayTimerRemain   : -
Segment-List Count : 1
Segment-List       : list1
Segment-List ID    : 1
List State         : Up
Verification State : -
PMTU               : 9600
Weight            : 1
Loop Detection State : Up
Network Slice ID   : -
Binding SID        : -
Reverse Binding ID : -
SID :
    2001:DB8:120::20
    2001:DB8:130::30
Binding SID        : 2001:DB8:100::450
Policy Name        : policy1
Path Verification  : Disable
Network Slice ID   : -
XclIndex           : 1
DelayTimerRemain   : -
SuppressTimeRemain : -
Active PMTU        : 9600
BFD State          : -
BFD Delay Remain   : -

```

## 步骤8 配置APN6实例

# 配置PE1。

```

[~PE1] apn
[*PE1-apn] ipv6
[*PE1-apn-ipv6] apn-id template tmplt1 length 64 app-group 32 user-group 32
[*PE1-apn-ipv6-template-tmplt1] app-group index 1 app-group1 length 10
[*PE1-apn-ipv6-template-tmplt1] user-group index 1 user-group1 length 8
[*PE1-apn-ipv6-template-tmplt1] quit
[*PE1-apn-ipv6] apn-id instance inst1
[*PE1-apn-ipv6-instance-inst1] template tmplt1
[*PE1-apn-ipv6-instance-inst1] apn-field app-group1 301
[*PE1-apn-ipv6-instance-inst1] apn-field user-group1 201
[*PE1-apn-ipv6-instance-inst1] quit
[*PE1-apn-ipv6] quit
[*PE1-apn] quit
[*PE1] commit

```

# 配置PE2。

```

[~PE2] apn
[*PE2-apn] ipv6
[*PE2-apn-ipv6] apn-id template tmplt1 length 64 app-group 32 user-group 32
[*PE2-apn-ipv6-template-tmplt1] app-group index 1 app-group1 length 10
[*PE2-apn-ipv6-template-tmplt1] user-group index 1 user-group1 length 8
[*PE2-apn-ipv6-template-tmplt1] quit
[*PE2-apn-ipv6] apn-id instance inst1
[*PE2-apn-ipv6-instance-inst1] template tmplt1
[*PE2-apn-ipv6-instance-inst1] apn-field app-group1 301
[*PE2-apn-ipv6-instance-inst1] apn-field user-group1 201
[*PE2-apn-ipv6-instance-inst1] quit
[*PE2-apn-ipv6] quit
[*PE2-apn] quit
[*PE2] commit

```

执行命令**display apn-id-ipv6 instance**查看APN6实例的相关信息。

以PE1的显示为例：

```

[~PE1] display apn-id-ipv6 instance name inst1
Instance Name      : inst1
Instance ID        : 1          APN ID Length : 64
APN ID             : 0x4b400000 0xc9000000
APN Mask           : 0xffc00000 0xff000000

```

## 步骤9 配置并应用APN ID隔离策略。

# 配置PE1。

```

[~PE1] apn
[*PE1-apn] ipv6

```

```
[*PE1-apn-ipv6] isolate-group name group10
[*PE1-apn-ipv6] apn-id isolate policy policy1
[*PE1-apn-ipv6-isolate-policy-policy1] index 1 instance inst1 isolate-group group10 behavior deny
[*PE1-apn-ipv6-isolate-policy-policy1] quit
[*PE1-apn-ipv6] isolate-group mapping-vpn
[*PE1-apn-ipv6-isolate-group-mapping-vpn] vpn-instance vpna peer-locator 2001:DB8:130:: 64 match
isolate-group group10
[*PE1-apn-ipv6-isolate-group-mapping-vpn] quit
[*PE1-apn-ipv6] quit
[*PE1-apn] quit
[*PE1] ip vpn-instance vpna
[*PE1-vpn-instance-vpna] apn-id-ipv6 isolate-policy policy1 inbound
[*PE1-vpn-instance-vpna] quit
[*PE1] commit
```

# 配置PE2。

```
[~PE2] apn
[*PE2-apn] ipv6
[*PE2-apn-ipv6] isolate-group name group10
[*PE2-apn-ipv6] apn-id isolate policy policy1
[*PE2-apn-ipv6-isolate-policy-policy1] index 1 instance inst1 isolate-group group10 behavior deny
[*PE2-apn-ipv6-isolate-policy-policy1] quit
[*PE2-apn-ipv6] isolate-group mapping-vpn
[*PE2-apn-ipv6-isolate-group-mapping-vpn] vpn-instance vpna peer-locator 2001:DB8:100:: 64 match
isolate-group group10
[*PE2-apn-ipv6-isolate-group-mapping-vpn] quit
[*PE2-apn-ipv6] quit
[*PE2-apn] quit
[*PE2] ip vpn-instance vpna
[*PE2-vpn-instance-vpna] apn-id-ipv6 isolate-policy policy1 inbound
[*PE2-vpn-instance-vpna] quit
[*PE2] commit
```

**步骤10** 配置业务流的过滤策略，符合条件的业务流按照指定的方式生成APN ID信息。如果端侧设备已经在报文中封装了APN ID，则此步骤可以省略。

# 配置PE1。

```
[~PE1] acl number 3333
[*PE1-acl-advance-3333] rule 5 permit ip source 11.11.11.11 0 destination 22.22.22.22 0
[*PE1-acl-advance-3333] commit
[~PE1-acl-advance-3333] quit
[~PE1] traffic classifier c1
[*PE1-classifier-c1] if-match acl 3333
[*PE1-classifier-c1] commit
[~PE1-classifier-c1] quit
[~PE1] traffic behavior b1
[*PE1-behavior-b1] remark apn-id-ipv6 instance inst1
[*PE1-behavior-b1] commit
[~PE1-behavior-b1] quit
[~PE1] traffic policy p1
[*PE1-trafficpolicy-p1] classifier c1 behavior b1
[*PE1-trafficpolicy-p1] share-mode
[*PE1-trafficpolicy-p1] statistics enable
[*PE1-trafficpolicy-p1] quit
[*PE1] interface GigabitEthernet 2/0/0
[*PE1-GigabitEthernet2/0/0] traffic-policy p1 inbound
[*PE1-GigabitEthernet2/0/0] commit
[~PE1-GigabitEthernet2/0/0] quit
[~PE1] interface GigabitEthernet 2/0/1
[*PE1-GigabitEthernet2/0/1] traffic-policy p1 inbound
[*PE1-GigabitEthernet2/0/1] commit
[~PE1-GigabitEthernet2/0/1] quit
[~PE1] interface GigabitEthernet 2/0/2
[*PE1-GigabitEthernet2/0/2] traffic-policy p1 inbound
[*PE1-GigabitEthernet2/0/2] commit
[~PE1-GigabitEthernet2/0/2] quit
```

# 配置PE2。

```
[~PE2] acl number 3333
[*PE2-acl-advance-3333] rule 5 permit ip source 22.22.22.22 0 destination 11.11.11.11 0
[*PE2-acl-advance-3333] commit
[~PE2-acl-advance-3333] quit
[~PE2] traffic classifier c1
[*PE2-classifier-c1] if-match acl 3333
[*PE2-classifier-c1] commit
[~PE2-classifier-c1] quit
[~PE2] traffic behavior b1
[*PE2-behavior-b1] remark apn-id-ipv6 instance inst1
[*PE2-behavior-b1] commit
[~PE2-behavior-b1] quit
[~PE2] traffic policy p1
[*PE2-trafficpolicy-p1] classifier c1 behavior b1
[*PE2-trafficpolicy-p1] share-mode
[*PE2-trafficpolicy-p1] statistics enable
[*PE2-trafficpolicy-p1] quit
[~PE2] interface GigabitEthernet 2/0/0
[*PE2-GigabitEthernet2/0/0] traffic-policy p1 inbound
[*PE2-GigabitEthernet2/0/0] commit
[~PE2-GigabitEthernet2/0/0] quit
[~PE2] interface GigabitEthernet 2/0/1
[~PE2-GigabitEthernet2/0/1] traffic-policy p1 inbound
[*PE2-GigabitEthernet2/0/1] commit
[~PE2-GigabitEthernet2/0/1] quit
[~PE2] interface GigabitEthernet 2/0/2
[~PE2-GigabitEthernet2/0/2] traffic-policy p1 inbound
[*PE2-GigabitEthernet2/0/2] commit
[~PE2-GigabitEthernet2/0/2] quit
```

#### 步骤11 配置隧道策略，引入私网流量

# 配置PE1。

```
[~PE1] route-policy p1 permit node 10
[*PE1-route-policy] apply extcommunity color 0:101
[*PE1-route-policy] quit
[*PE1] bgp 100
[*PE1-bgp] l2vpn-family evpn
[*PE1-bgp-af-evpn] peer 2001:DB8:3::3 route-policy p1 import
[*PE1-bgp-af-evpn] quit
[*PE1-bgp] quit
[*PE1] tunnel-policy p1
[*PE1-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
[*PE1-tunnel-policy-p1] quit
[*PE1] ip vpn-instance vpna
[*PE1-vpn-instance-vpna] ipv6-family
[*PE1-vpn-instance-vpna-af-ipv6] tn1-policy p1 evpn
[*PE1-vpn-instance-vpna-af-ipv6] commit
[~PE1-vpn-instance-vpna-af-ipv6] quit
[~PE1-vpn-instance-vpna] quit
```

# 配置PE2。

```
[~PE2] route-policy p1 permit node 10
[*PE2-route-policy] apply extcommunity color 0:101
[*PE2-route-policy] quit
[*PE2] bgp 100
[*PE2-bgp] l2vpn-family evpn
[*PE2-bgp-af-evpn] peer 2001:DB8:1::1 route-policy p1 import
[*PE2-bgp-af-evpn] quit
[*PE2-bgp] quit
[*PE2] tunnel-policy p1
[*PE2-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
[*PE2-tunnel-policy-p1] quit
[*PE2] ip vpn-instance vpna
[*PE2-vpn-instance-vpna] ipv6-family
[*PE2-vpn-instance-vpna-af-ipv6] tn1-policy p1 evpn
[*PE2-vpn-instance-vpna-af-ipv6] commit
```

```
[~PE2-vpn-instance-vpna-af-ipv6] quit
[~PE2-vpn-instance-vpna] quit
```

配置完成后，执行命令**display ipv6 routing-table vpn-instance vpna**查看VPN实例路由表信息，可以看到私网路由已经成功迭代到SRv6 TE Policy。

以PE1的显示为例：

```
[~PE1] display ipv6 routing-table vpn-instance vpna
Routing Table : vpna
Destinations : 6      Routes : 6

Destination : 2001:DB8:111::111      PrefixLength : 128
NextHop     : 2001:DB8:11::2          Preference   : 255
Cost        : 0                      Protocol     : EBGp
RelayNextHop : 2001:DB8:11::2        TunnelID    : 0x0
Interface   : GigabitEthernet2/0/0   Flags       : RD

Destination : 2001:DB8:222::222      PrefixLength : 128
NextHop     : 2001:DB8:3::3          Preference   : 255
Cost        : 0                      Protocol     : IBGP
RelayNextHop : ::                   TunnelID    : 0x000000003400000001
Interface   : policy1               Flags       : RD

Destination : 2001:DB8:11::          PrefixLength : 64
NextHop     : 2001:DB8:11::1          Preference   : 0
Cost        : 0                      Protocol     : Direct
RelayNextHop : ::                   TunnelID    : 0x0
Interface   : GigabitEthernet2/0/0   Flags       : D

Destination : 2001:DB8:11::1          PrefixLength : 128
NextHop     : ::1                    Preference   : 0
Cost        : 0                      Protocol     : Direct
RelayNextHop : ::                   TunnelID    : 0x0
Interface   : GigabitEthernet2/0/0   Flags       : D

Destination : 2001:DB8:22::          PrefixLength : 64
NextHop     : 2001:DB8:3::3          Preference   : 255
Cost        : 0                      Protocol     : IBGP
RelayNextHop : ::                   TunnelID    : 0x000000003400000001
Interface   : policy1               Flags       : RD

Destination : FE80::                 PrefixLength : 10
NextHop     : ::                     Preference   : 0
Cost        : 0                      Protocol     : Direct
RelayNextHop : ::                   TunnelID    : 0x0
Interface   : NULL0                 Flags       : DB

[~PE1] display ipv6 routing-table vpn-instance vpna 2001:DB8:222::222 verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
-----
Routing Table : vpna
Summary Count : 1

Destination : 2001:DB8:222::222      PrefixLength : 128
NextHop     : 2001:DB8:3::3          Preference   : 255
Neighbour   : 2001:DB8:3::3          ProcessID    : 0
Label       : NULL                   Protocol     : IBGP
State       : Active Adv Relied      Cost         : 0
Entry ID    : 0                      EntryFlags   : 0x00000000
Reference Cnt: 0                     Tag          : 0
Priority     : low                    Age          : 1867sec
IndirectID  : 0x10000DC               Instance     :
RelayNextHop : ::                   TunnelID    : 0x000000003400000001
Interface   : policy1               Flags       : RD
```

## 步骤12 检查配置结果

同一VPN的CE能够相互Ping通，例如：

```
[~CE1] ping ipv6 -a 2001:DB8:111::111 2001:DB8:222::222
PING 2001:DB8:222::222 : 56 data bytes, press CTRL_C to break
Reply from 2001:DB8:222::222
bytes=56 Sequence=1 hop limit=62 time=148 ms
Reply from 2001:DB8:222::222
bytes=56 Sequence=2 hop limit=62 time=31 ms
Reply from 2001:DB8:222::222
bytes=56 Sequence=3 hop limit=62 time=46 ms
Reply from 2001:DB8:222::222
bytes=56 Sequence=4 hop limit=62 time=28 ms
Reply from 2001:DB8:222::222
bytes=56 Sequence=5 hop limit=62 time=26 ms

--- 2001:DB8:222::222 ping statistics---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
```

----结束

## 配置文件

- PE1的配置文件

```
sysname PE1
#
apn
ipv6
  apn-id template tmplt1 length 64 app-group 32 user-group 32
  app-group index 1 app-group1 length 10
  user-group index 1 user-group1 length 8
  apn-id instance inst1
  template tmplt1
  apn-field app-group1 301
  apn-field user-group1 201
  isolate-group name group10
  apn-id isolate policy policy1
  statistics enable
  index 1 instance inst1 isolate-group group10 behavior deny
  isolate-group mapping-vpn
  vpn-instance vpna peer-locator 2001:DB8:130:: 64 match isolate-group group10
#
acl number 3333
rule 5 permit ip source 11.11.11.11 0 destination 22.22.22.22 0
#
traffic classifier c1
if-match acl 3333
#
traffic behavior b1
remark apn-id-ipv6 instance inst1
#
traffic policy p1
share-mode
statistics enable
classifier c1 behavior b1
#
ip vpn-instance vpna
  ipv6-family
    route-distinguisher 100:1
    apply-label per-instance
    vpn-target 1:1 export-extcommunity evpn
    vpn-target 1:1 import-extcommunity evpn
    tn1-policy p1 evpn
  apn-id-ipv6 isolate-policy policy1 inbound
#
segment-routing ipv6
  encapsulation source-address 2001:DB8:1::1
  locator PE1 ipv6-prefix 2001:DB8:100:: 64 static 32
  opcode ::10 end psp
```



```
srv6-te-policy locator PE1
segment-list list1
index 5 sid ipv6 2001:DB8:120::20
index 10 sid ipv6 2001:DB8:130::30
srv6-te policy policy1 endpoint 2001:DB8:3::3 color 101
binding-sid 2001:DB8:100::450
candidate-path preference 100
segment-list list1
mapping-policy p1 color 101
match-type apn-id-ipv6
index 1 instance inst1 match srv6-te-policy color 101
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0001.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator PE1 auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:10::1/64
isis ipv6 enable 1
undo dcn
undo dcn mode vlan
#
interface GigabitEthernet2/0/0
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001:DB8:11::1/64
traffic-policy p1 inbound
undo dcn
undo dcn mode vlan
#
interface GigabitEthernet2/0/1
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001:DB8:12::1/64
undo dcn
undo dcn mode vlan
traffic-policy p1 inbound
#
interface GigabitEthernet2/0/2
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001:DB8:13::1/64
undo dcn
undo dcn mode vlan
traffic-policy p1 inbound
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:1::1/128
isis ipv6 enable 1
#
bgp 100
router-id 1.1.1.1
private-4-byte-as enable
peer 2001:DB8:3::3 as-number 100
peer 2001:DB8:3::3 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
```

```
#
ipv6-family vpn-instance vpna
import-route direct
advertise l2vpn evpn
segment-routing ipv6 locator PE1 evpn
segment-routing ipv6 traffic-engineer best-effort evpn
peer 2001:DB8:11::2 as-number 65410
#
l2vpn-family evpn
policy vpn-target
peer 2001:DB8:3::3 enable
peer 2001:DB8:3::3 route-policy p1 import
peer 2001:DB8:3::3 advertise encap-type srv6
#
route-policy p1 permit node 10
apply extcommunity color 0:101
#
tunnel-policy p1
tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
#
return
```

- P的配置文件

```
sysname P
#
segment-routing ipv6
encapsulation source-address 2001:DB8:2::2
locator P ipv6-prefix 2001:DB8:120:: 64 static 32
opcode ::20 end psp
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0002.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator P auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:10::2/64
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:20::1/64
isis ipv6 enable 1
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:2::2/128
isis ipv6 enable 1
#
return
```

- PE2的配置文件

```
sysname PE2
#
apn
ipv6
apn-id template tmplt1 length 64 app-group 32 user-group 32
app-group index 1 app-group1 length 10
user-group index 1 user-group1 length 8
apn-id instance inst1
template tmplt1
apn-field app-group1 301
apn-field user-group1 201
```

```
isolate-group name group10
apn-id isolate policy policy1
statistics enable
index 1 instance inst1 isolate-group group10 behavior deny
isolate-group mapping-vpn
vpn-instance vpna peer-locator 2001:DB8:100:: 64 match isolate-group group10
#
acl number 3333
rule 5 permit ip source 22.22.22.22 0 destination 11.11.11.11 0
#
traffic classifier c1
if-match acl 3333
#
traffic behavior b1
remark apn-id-ipv6 instance inst1
#
traffic policy p1
share-mode
statistics enable
classifier c1 behavior b1
#
ip vpn-instance vpna
ipv6-family
route-distinguisher 200:1
apply-label per-instance
vpn-target 1:1 export-extcommunity evpn
vpn-target 1:1 import-extcommunity evpn
tnl-policy p1 evpn
apn-id-ipv6 isolate-policy policy1 inbound
#
segment-routing ipv6
encapsulation source-address 2001:DB8:3::3
locator PE2 ipv6-prefix 2001:DB8:130:: 64 static 32
opcode ::30 end psp
srv6-te-policy locator PE2
segment-list list1
index 5 sid ipv6 2001:DB8:120::20
index 10 sid ipv6 2001:DB8:100::10
srv6-te policy policy1 endpoint 2001:DB8:1::1 color 101
binding-sid 2001:DB8:130::350
candidate-path preference 100
segment-list list1
#
isis 1
is-level level-1
cost-style wide
network-entity 10.0000.0000.0003.00
#
ipv6 enable topology ipv6
segment-routing ipv6 locator PE2 auto-sid-disable
#
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:20::2/64
isis ipv6 enable 1
#
interface GigabitEthernet2/0/0
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001:DB8:22::1/64
traffic-policy p1 inbound
#
interface GigabitEthernet2/0/1
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
```

```
ipv6 address 2001:DB8:23::1/64
undo dcn
undo dcn mode vlan
traffic-policy p1 inbound
#
interface GigabitEthernet2/0/2
undo shutdown
ip binding vpn-instance vpna
ipv6 enable
ipv6 address 2001:DB8:24::1/64
undo dcn
undo dcn mode vlan
traffic-policy p1 inbound
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:3::3/128
isis ipv6 enable 1
#
bgp 100
router-id 3.3.3.3
private-4-byte-as enable
peer 2001:DB8:1::1 as-number 100
peer 2001:DB8:1::1 connect-interface LoopBack1
#
ipv4-family unicast
undo synchronization
#
ipv6-family vpn-instance vpna
import-route direct
advertise l2vpn evpn
segment-routing ipv6 locator PE2 evpn
segment-routing ipv6 traffic-engineer best-effort evpn
peer 2001:DB8:22::2 as-number 65420
#
l2vpn-family evpn
policy vpn-target
peer 2001:DB8:1::1 enable
peer 2001:DB8:1::1 route-policy p1 import
peer 2001:DB8:1::1 advertise encap-type srv6
#
route-policy p1 permit node 10
apply extcommunity color 0:101
#
tunnel-policy p1
tunnel select-seq ipv6 srv6-te-flow-group load-balance-number 1
#
return
```

- CE1的配置文件

```
sysname CE1
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:11::2/64
#
interface GigabitEthernet2/0/1
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:12::2/64
#
interface GigabitEthernet2/0/2
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:13::2/64
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:111::111/128
```

```
#
bgp 65410
router-id 10.11.1.1
peer 2001:DB8:11::1 as-number 100
#
ipv4-family unicast
undo synchronization
#
ipv6-family unicast
undo synchronization
import-route direct
peer 2001:DB8:11::1 enable
#
return
```

- CE2的配置文件

```
sysname CE2
#
interface GigabitEthernet1/0/0
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:22::2/64
#
interface GigabitEthernet2/0/1
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:23::2/64
#
interface GigabitEthernet2/0/2
undo shutdown
ipv6 enable
ipv6 address 2001:DB8:24::2/64
#
interface LoopBack1
ipv6 enable
ipv6 address 2001:DB8:222::222/128
#
bgp 65420
router-id 10.22.2.2
peer 2001:DB8:22::1 as-number 100
#
ipv4-family unicast
undo synchronization
#
ipv6-family unicast
undo synchronization
import-route direct
peer 2001:DB8:22::1 enable
#
return
```