

# IPv6

(提示: 由于内容较多, 阅读时, 建议开启 文档结构图.)

作者: 日月同辉

## 目录

概述 .....	1
IPv6 地址格式.....	2
IPv6 地址表示方法.....	2
IPv6 地址类型.....	10
配置IPv6 地址.....	14
IPv6 静态路由.....	17
IPv6 静态路由配置实验.....	18
IPv6 RIP (RIPng).....	22
IPv6 OSPF (OSPFv3) .....	30
IPv6 EIGRP (EIGRP v6) .....	43
IPv6 BGP.....	50
IPv6 隧道.....	62
IPv6 组播.....	79

## 概述

在我们现有的网络中, 几乎所有网络都使用 IP 协议作为通信的地址协议, 我们的网络使用 IP 来表示地址信息, 每一个节点都应该分配一个唯一的地址, 才能保证通信正常。现在正常使用的 IP 协议为版本 4, 用 32 位来表示, 地址空间为  $2^{32}$ , 结果约为 42.9 亿, 需要说明的是, 虽然地址共有 42.9 亿之多, 但并不表示这些地址可以供 42.9 亿个节点使用, 因为我们的地址是分网段的, 也就是说即使在一个节点的情况下, 分配地址时, 也是分配一个网段而不是一个地址, 所以这样就使得版本 4 的 IP 地址一下子变得空间狭小, 再加之有相当一部分地址是不可用的, 那么随着网络的迅速膨胀, IP ver4 的地址空间变得几乎快耗尽了。在这样的情况下, 出现了一些如 VLSM 子网技术, NAT 网络地址翻译技术, 试图来缓和地址空间的快速消耗。与此同时, 人们也开发出了一个地址空间更为庞大的 IP 协议, 这个协议拥有比 IP ver4 多出数倍的地址空间, 来解决网络地址匮乏的问题, 这个 IP 协议就是 IP 版本 6, 即 IPv6。

## IPv6 地址格式

IPv6 拥有更为庞大的地址空间，是因为 IPv4 只是采用 32 位来表示，而 IPv6 采用 128 位来表示，这样大的一个地址空间，几乎可以容纳无数个节点。正因为 IPv6 使用了 128 位来表示地址，在表示和书写上面具有相当的困难，原来的 IPv4 使用 10 进制来表示，而 IPv6 由于地址太长，则采用 16 进制来表示，但无论我们如何表示，计算机都是处理二进制。因为 10 进制表示时，使用 0 到 9 共十个数字来表示，而 16 进制需要在 10 进制原有的基础上多出 6 个数字，即需要多出 11,12,13,14,15，这 6 个数字则采用字母的形式来表示，分别为

A(表示 10)，B(表示 11)，C(表示 12)，D(表示 13)，E(表示 14)，F(表示 15)，这些字母是不区别大小写的。

但是由于 IPv6 拥有 128 位的长度，所以不能直接表示，必须像 IPv4 那样进行分段表示。IPv6 将整个地址分为 8 段来表示，每段之间用冒号隔开，每段的长度为 16 位，表示如下：

**XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX**

从上面可以看出，IPv6 中每一个段是 16 位，每段共四个 X，其中 X 使用 4 bit 表示，一个 X 就表示一个数字或字母，一个完整的地址共 128 bit。

一个 X 使用 4 bit 表示，那么 XXXX 的取值范围就应该从 0000 到 FFFF。

## IPv6 地址表示方法

对于一个完整的 IPv6 地址，需要写 128 位，已经被分成了 8 段，每段 4 个字符，也就是说完整地表示一个 IPv6 地址，需要写 32 个字母，这是相当长的，并且容易混淆和出错，所以 IPv6 在地址的表示方法上，是有讲究的，到目前为止，IPv6 地址的表示方法分为三种，分别是：

### 1. 首选格式

### 2. 压缩表示

### 3. IPv4 内嵌在 IPv6 中

下面分别详细介绍这三种 IPv6 地址表示方法：

#### 1. 首选格式

首选格式的表示方法其实没有任何讲究，就是将 IPv6 中的 128 位，也就是共 32 个字符完完整整，一个不漏地全写出来，比如下面就是一些 IPv6 地址的首选格式表示形式：

**0000:0000:0000:0000:0000:0000:0000:0000**

**0000:0000:0000:0000:0000:0000:0000:0001**

**2001:0410:0000:1234:FB00:1400:5000:45FF**

**3ffe:0000:0000:0000:1010:2a2a:0000:0001**

**FE80:0000:0000:0000:0000:0000:0000:0009**

**FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**

从上面 IPv6 地址的首选格式表示中可以看出，每一个地址，都将 32 个字符全部写了出来，即使地址中有许多个 0，或者有许多个 F，也都一个不漏地写了出来，由此可见，首选格式只需要将地址完整写出即可，没有任何复杂的变化，但是容易出错。

## 2. 压缩格式

从前面一个 IPv6 地址表示方法首选格式表示方法中可以看出，一个完整的 IPv6 地址中，会经常性的出现许多个 0，而我们知道，许多时候，0 是毫无意义的，0 表示没有，写出来，也表示没有，不写，也同样表示没有，那么我们就考虑能否将不影响地址结果的 0 给省略不写，这样就可以大大节省时间，也方便人们阅读和书写，这样的将地址省略 0 的表示方法，称为压缩格式。

而压缩格式的表示中，分三种情况，下面来分别介绍三种压缩格式：

### 第一种情况：

在 IPv6 中，地址分为 8 个段来表示，每个段共 4 个字符，但是一个完整的 IPv6 地址会经常碰到整个段 4 个字符全部都为 0，所以我们将整个段 4 个字符全部都为 0 的使用双冒号 :: 来表示，如果连续多个段全部都为 0，那么也可以同样将多个段都使用 双冒号 :: 来表示，如果是多个段，并不需要将双冒号写多次，只需要写一次即可，比如一个地址 8 个段，其中有三个段全部都为 0，那么我们就将这全为 0 的三个段共 48 位用 :: 来表示，再将其它 5 个段照常写出即可，当计算机读到这样一个不足 128 位的地址时，比 128 位少了多少位，就在 :: 的地方补上多少个 0，比如上面的 :: 代替为 48 位，那么计算机就会在这个地址的 :: 位置补上 48 位的 0，这样就正确地将地址还原回去了。

下面来看一些整个段 4 个字符都为 0 的 IPv6 地址使用压缩格式来表示：

### 例 1：

压缩前：

**0000:0000:0000:0000:0000:0000:0000:0000**

压缩后:

**::**

**说明:** 可以看出, 由于这个地址的 8 个段全部都为 0, 所以只用**::**就将整个地址表示出来, 当计算机拿到这个压缩后的地址时, 发现比正常的 128 位少了 128 位, 那么就会在**::**的地方补上 128 个 0, 结果为:

**0000:0000:0000:0000:0000:0000:0000:0000**

可以看出, 计算机还原的地址就是压缩之前的真实地址。

**例 2:**

压缩前:

**0000:0000:0000:0000:0000:0000:0000:0001**

压缩后:

**::0001**

**说明:** 可以看出, 压缩后的地址比正常的 128 位少了 112 位, 计算机就会在**::**的地方补上 112 个 0, 结果为:

**0000:0000:0000:0000:0000:0000:0000:0001**

可以看出, 计算机还原的地址就是压缩之前的真实地址。

**例 3:**

压缩前:

**2001:0410:0000: 0000:FB00:1400:5000:45FF**

压缩后:

**2001:0410 :: FB00:1400:5000:45FF**

**说明:** 可以看出, 压缩后的地址比正常的 128 位少了 32 位, 计算机就会在::的地方补上 32 个 0, 结果为:

**2001:0410:0000: 0000:FB00:1400:5000:45FF**

可以看出, 计算机还原的地址就是压缩之前的真实地址。

例 4:

压缩前:

**3ffe:0000:0000:0000:1010:2a2a:0000:0001**

压缩后:

**3ffe::1010:2a2a::0001**

**说明:** 当计算机拿到这个压缩后的地址, 发现比正常的 128 位少了 64 位, 计算机就会试图在::的地方补上少了的 64 个 0, 但是我们可以看到, 压缩后的地址有两个::, 而计算机要补上 64 个 0, 所以这时补出来的结果很可能是以下几种:

**3ffe:0000:1010:2a2a: 0000:0000:0000:0001**

或

**3ffe:0000:00001010:2a2a::0000:0000:0001**

或

**3ffe:0000:0000:0000:1010:2a2a:0000:0001**

从结果中可以发现，当一个 IPv6 地址被压缩后，如果计算机出现两个或多个 :: 的时候，计算机在将地址还原时，就可能出现多种情况，这将导致计算机还原后的地址不是压缩之前的地址，将导致地址错误，最终通信失败。

所以，在压缩 IPv6 地址时，一个地址中只能出现一个 ::。

## 第二种情况：

在压缩格式的第一种情况的表示中，是在地址中整个段 4 个字符都为 0 时，才将其压缩为 :: 来表示，但是在使用第一种情况压缩之后，我们仍然可以看见地址中还存在许多毫无意义的 0，比如 0001，0410。我们知道，0001 中，虽然前面有三个 0，但是如果我们将前面的 0 全部省略掉，写为 1，结果是等于 0001 的，而 0410 也是一样，我们将前面的 0 省略掉，写成 410，也同样等于 0410 的，所以我们在省略数字前面的 0 时，是不影响结果的，那么这个时候，表示 IPv6 地址时，允许将一个段中前导部分的 0 省略不写，因为不影响结果。但是需要注意的是，如果 0 不是前导 0，比如 2001，我们就不能省略 0 写成 21，因为 21 不等于 2001，所以在中间的 0 不能省略，只能省略最前面的 0。下面来看一些省略前导 0 的地址表示形式：

### 例 1：

压缩前：

**0000:0000:0000:0000:0000:0000:0000:0000**

压缩后：

**0:0:0:0:0:0:0:0**

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

例 2:

压缩前:

**0000:0000:0000:0000:0000:0000:0000:0001**

压缩后:

**0:0:0:0:0:0:0:1**

从结果中可以看出, 计算机根本就不需要对这样的地址还原, 压缩后的结果和压缩前的结果是相等的。

例 3:

压缩前:

**2001:0410:0000:1234:FB00:1400:5000:45FF**

压缩后:

**2001:410:0:1234:FB00:1400:5000:45FF**

从结果中可以看出, 计算机根本就不需要对这样的地址还原, 压缩后的结果和压缩前的结果是相等的。

第三种情况:

在前面两种 IPv6 地址的压缩表示方法中, 第一种是在整段 4 个字符全为 0 时, 才将其压缩后写为 ::, 而第二种是将无意义的 0 省略不写, 可以发现两种方法都能节省



时间，方便阅读。第三种压缩方法就是结合前两种方法，既将整段 4 个字符全为 0 的部分写成 **::**，也将无意义的 0 省略不写，结果就可以出现以下一些最方便的表示方法：

例 1:

压缩前:

**0000:0000:0000:0000:0000:0000:0000:0001**

压缩后:

**:: 1**

可以看到，结合了两种压缩格式的方法，但为简便。

例 2:

压缩前:

**2001:0410:0000:0000:FB00:1400:5000:45FF**

压缩后:

**2001:410:: FB00:1400:5000:45FF**

可以看到，结合了两种压缩格式的方法，但为简便。

### 3.IPv4 内嵌在 IPv6 中

在网络还没有全部从 IPv4 过渡到 IPv6 时，就可能出现某些设备即连接了 IPv4 网络，又连接了 IPv6 网络，对于这样的情况，就需要一个地址即可以表示 IPv4 地址，

又可以表示 IPv6 地址。

因为一个 IPv4 地址为 32 位，一个 IPv6 地址为 128 位，要让一个 IPv4 地址表示为 IPv6 地址，明显已经少了 96 位，那么就将一个正常的 IPv4 地址通过增加 96 位，结果变成 128 位，来与 IPv6 通信。在表示时，是在 IPv4 原有地址的基础上，增加 96 个 0，结果变成 128 位，增加的 96 个 0 再结合原有的 IPv4 地址，表示方法为

**0:0:0:0:0:A.B.C.D 或者 ::A.B.C.D.**，如下：

**0000: 0000: 0000: 0000: 0000:0000:A.B.C.D**

96 个 0          32 位

例：

IPv4 地址为 138.1.1.1

表示 IPv6 地址为 0:0:0:0:0:138.1.1.1

**注：**IPv6 中没有广播地址，IPv6 不建议划子网，如果需要划子网，网络位请不要低于 48 位。

## IPv6 地址类型

在 IPv4 地址中，地址分许多类型，比如代表节点自己的 127.0.0.0/8,私有地址段，组播地址段，广播地址，以及一些不可用的地址。在 IPv6 中，同样地址也像 IPv4 那样分了许多类型，我们需要了解的有 3 种类型，为 Unicast（单播），Anycast（任意播）和 Multicast（组播），下面分别来详细介绍这几种地址类型。

### Unicast（单播）

即使是在 IPv4 中，单播地址的类型也分好多种，就是我们常用的也分私有，公有，还有回环地址，在 IPv6 中，单播地址也分好几种，我们需要知道的有：

Link-Local Address（链路本地地址），Unique Local Address（本地站点地址），Aggregatable Global Address（可聚合全球），回环地址。

下面详细介绍几种单播地址：

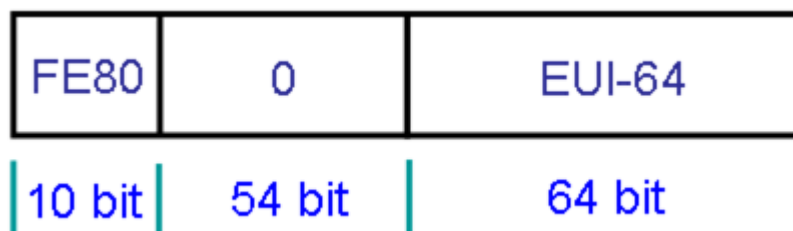
### Link-Local Address（链路本地地址）

即使网络再大，每两点之间，都有链路相连，在一个节点将数据包发给下一个节点时，必须在数据包中封装三层 IP 地址，再封装下一节点的二层链路地址（如以太网中的 MAC 地址），才能将数据包发给下一节点，并且只有当封装的二层链路地址确实为下一节点的真实链路地址时，对方才能接收，这就是普通二层链路地址的功能，这样的地址在一条链路的范围内明确了每个节点，并且这样的地址是不能被路由的。

而在 IPv6 网络中，两个 IPv6 的节点通过链路相连，必须在这条链路之间为各自确立一个 Link-Local Address（即链路本地地址），在一条链路上，IPv6 节点能够确定对方节点的身份，能够将数据包发向对方节点，必须知道对方节点的链路本地地址，如果不知道，将是不能通信的，所以一条链路中的 IPv6 节点要通信，必须拥有链路本地地址，并且这个链路本地地址只在一条链路中有效，也不能被路由，而不同链路的链路本地地址是可以重复的。

因为在链路上没有链路本地地址的情况下，IPv6 是不能通信的，所以每个节点必须拥有一个链路本地地址，当一个节点上正常启动了 IPv6 之后，链路本地地址是不需要人工干预，会自己生成的，但也可以自己手工配置链路本地地址。

自动生成的链路本地地址，有默认的特殊格式，是以 FE80::/10 (1111 1110 10) 打头，再加 54 个 0，还差 64 位，这后面的 64 位，再使用 EUI-64 来填充，表示如下：



### EUI-64 结构

一个链路本地地址的后 64 位使用 EUI-64 来填充，EUI-64 其实就是接口的 MAC 地址，而 MAC 地址共长度为 48 位，要填充 64 位的 EUI-64，还少 16 位。一个完整的 EUI-64 是将 MAC 地址的 48 位平均分成两部分，前面 24 位，后面 24 位，然后在中间补上 FFFE(16 位)，如一个 MAC 地址为 00:12:33:5C:82:E1，将其变为 EUI-64 的结果如下表示：



只有以太网链路才会有 MAC 地址，而串行链路是没有 MAC 地址的，当一个接口上启用 IPv6 之后，此接口会自动产生一个链路本地地址，而链路本地地址需要借用接口上的 MAC 地址才能产生，这在有 MAC 地址的以太网接口上可以轻松实现，但是当一个没有 MAC 地址的串行接口上开启 IPv6 之后，由于自己没有 MAC 地址，所以不能产生 EUI-64，也就无法完成链路本地地址。在这种情况下，所有没有 MAC 地址的接口，如串行接口，在开启 IPv6 后，需要产生 EUI-64 时，统统借用设备上第一个以太网插槽的第一个接口，也可以理解为没有 MAC 地址的接口，统统使用设备上 MAC 地址池中的第一个地址，比如设备上为接口 s2/3 开启 IPv6 之后，就很有可能借用 F0/0 或 G0/0 接口的 MAC 地址。

EUI-64 不仅在产生链路本地地址时可以使用，在正常配置 IPv6 地址时，同样可以使用 EUI-64 来填充后 64 位。

### 本地站点地址

本地站点地址是单播中一种受限制的地址，只在一个站点内使用，不会默认启用，这个地址不能在公网上路由，只能在一个指定的范围内路由，需要手工配置。IPv6 中的本地站点地址类似 IPv4 中私有地址，如 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16。

得不到合法 IPv6 地址的机构可配置本地站点地址，表示方法为：

FC00::/7 + 41bit 子网标识 +16bit 链路标识+ EUI-64

## 可聚合全球单播地址

可聚合全球单播地址相当于 IPv4 的公网地址，可以被路由的，可以正常使用的地址，但网络位最少为 48 位。可聚合全球单播地址的范围是

2000:0000:0000:0000:0000:0000:0000:0000

到

3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

以上面可以看出，可聚合全球单播地址也就是 2 和 3 打头的地址，因为 IPv6 使用 16 进制来表示，一个字符的取值范围从 0 到 F 共 16 个，而可聚合全球单播地址地址占了 2 和 3 两个，由此说明，可聚合全球单播地址占 IPv6 总地址空间的 8 分之 1，也就是说，所有 IPv6 地址中，只有 8 分之 1 是可以给网络正常使用的。

## 回环地址

回环地址表示节点自身，类似 IPv4 的 127.0.0.0/8

回环地址表示为

0000:0000:0000:0000:0000:0000:0000:0001

0:0:0:0:0:0:0:1

::1

## 任意播地址

任意播地址表示一组接口，当一个发向某个任意播地址的数据包，只被最近的接口收到，这个地址是由路由协议定义的，不能手工配置，但是我们无法看到一个地址就能区别出到底是单播地址还是任意播地址，因为任意播地址的表示格式和单播地址是一样的，也就是说任意播地址就是用普通的单播地址来表示的。任意播地址只能出现在路由器上，并且不能作为数据包的源地址来使用。

## 组播

组播地址就是一个目标为组播地址的数据包将被多个节点收到，地址以 FF00::/8 (1111 1111)打头，.表示为

FF00:0000:0000:0000:0000:0000:0000/8

FF00:0:0:0:0:0:0:0/8

FF00::/8

详细的 IPv6 多播知识将在后面的 IPv6 Multicast 部分介绍。

## 配置IPv6 地址

### 1.激活 IPv6 功能

默认情况下，Cisco 设备的 IPv6 流量转发功能是关闭的，需要使用 IPv6，必须先开启 IPv6 流量转发功能。

#### (1) 开启 IPv6 流量转发功能

```
r1(config)#ipv6 unicast-routing
```

### 2.配置正常的 IPv6 地址

#### (1) 在接口下配置正常 IPv6 地址

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2011:1:2:3:1:1:1:1/64
```

**说明：**配置的地址前 64 位为网络地址，即 2011:1:2:3；后 64 位为主机位，即 1:1:1:1。

#### (2) 查看接口的 IPv6 地址

```
r1#show ipv6 interface brief f0/0
```

```
FastEthernet0/0      [up/up]
```

```
FE80::C200:EFF:FEB0:0
```

```
2011:1:2:3:1:1:1:1
```

```
r1# r1#
```

**说明：**可以看到接口 F0/0 已经接受我们配置的地址 2011:1:2:3:1:1:1:1。

### 3.使用 EUI-64 格式配置静态地址：

#### (1)配置包含 EUI-64 的 IPv6 地址

```
r1(config)#int f0/1
```

```
r1(config-if)#ipv6 address 2022:2:2:22::/64 eui-64
```

#### (2) 查看接口的 IPv6 地址

```
r1#show interfaces f0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
Hardware is Gt96k FE, address is c000.0eb0.0000 (bia c000.0eb0.0000)
```

```
r1#show ipv6 interface brief f0/1
```

```
FastEthernet0/1      [up/up]
```

```
FE80::C200:EFF:FEB0:1
```

```
2022:2:2:22:C200:EFF:FEB0:1
```

```
r1#
```

**说明：**可以看到，F0/1 成功使用接口上的 MAC 地址为 EUI-64 来填充后 64 位。

#### 4.仅启用接口 IPv6 功能

**说明：**一个接口上可以仅启用 IPv6 功能，而不配置 IPv6 地址

##### （1）启用接口 IPv6 功能

```
r1(config)#int s1/0
```

```
r1(config-if)#ipv6 enable
```

##### （2）查看接口 IPv6 状态

```
r1#show ipv6 interface brief serial 1/0
```

```
Serial1/0          [up/up]
```

```
FE80::C200:EFF:FEB0:0
```

```
r1#
```

可以看到，接口 **S1/0** 可以只开启 IPv6 功能而不配地址，但开了 IPv6 功能的接口也会自动产生一个链路本地地址。

#### 5.配置无编号地址

当地址紧缺时，可以配置一个接口使用另外一个接口的地址，这样的地址称为无编号地址，即 **unnumbered** 地址，当从无编号接口产生数据包时，该接口使用借用的那个接口的地址作为源地址，配置这样的地址，需要允许双方不同网段协议的配合。

##### （1）为接口配置无编号地址

```
r1(config)#int s1/1
```

```
r1(config-if)#ipv6 unnumbered f0/0
```

##### （2）查看接口 IPv6 地址情况。

```
r1#show ipv6 interface brief serial 1/1
```



Serial1/1 [up/up]

FE80::C200:EFF:FEB0:0

unnumbered (FastEthernet0/0)

r1#

**说明：**可以看到结果显示为接口 S1/1 借用 F0/0 的地址。

## IPv6 静态路由

在 IPv6 中，静态路由的写法分三种，分别为：

### 1. 直连静态路由（Directly Attached Static Routes）

写法为只指定路由的出口，目标网络被认为是和此接口直连的，但此方法在接口为多路访问时，会有问题。

**例配：**

```
ipv6 route 2022:2:2:22::/64 s1/1
```

**说明：**到达目标网络 2022:2:2:22::/64 的数据包从接口 s1/1 发出去。

### 2. 递归静态路由（Recursive Static Routes）

写法为只指定路由的下一跳地址，此方法在任何网络环境中可行。

**例配：**

```
r1(config)#ipv6 route 2022:2:2:22::/64 2012:1:1:11::2
```

**说明：**到达目标网络 2022:2:2:22::/64 的数据包发给下一跳地址 2012:1:1:11::2。

### 3. 完全静态路由（Fully Specified Static Routes）

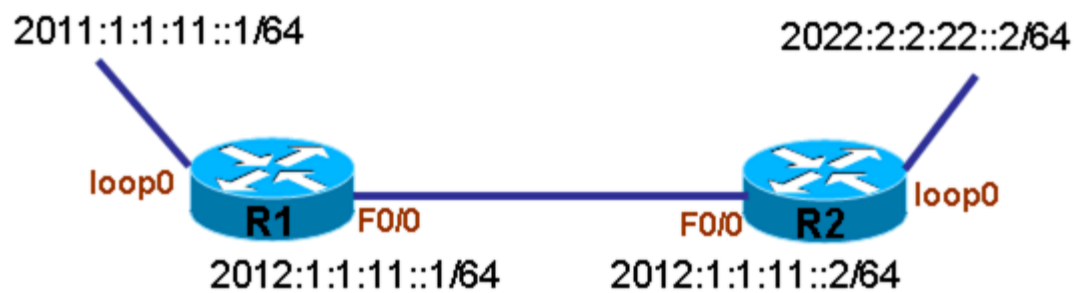
写法为同时指定出口和下一跳地址，只有当出口为多路访问时，并且确实需要明确指定下一跳时，才需要写完全静态路由，下一跳必须是和出口同网段的。

例配：

```
r1(config)#ipv6 route 2022:2:2:22::/64 f0/0 2012:1:1:11::2
```

**说明：**到达目标网络 2022:2:2:22::/64 的数据包从接口 F0/0 发出去，并且交给下一跳地址 2012:1:1:11::2。

## IPv6 静态路由配置实验



**说明：**配置静态路由，使双方都能 ping 通互相 loopback 接口的网段。

由于是多路访问接口，所以省去配置直连静态路由的方法。

### 1.网络初始配置：

#### (1) R1 初始配置：

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config)#int loopback 0
```

```
r1(config-if)#ipv6 address 2011:1:1:11::1/64
```

```
r1(config-if)#
```

## (2) R2 初始配置:

```
r2(config)#ipv unicast-routing
```

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv address 2012:1:1:11::2/64
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

```
r2(config-if)#
```

## 2.在 R1 上配置递归静态路由

### (1) 配置递归静态路由

```
r1(config)#ipv6 route 2022:2:2:22::/64 2012:1:1:11::2
```

**说明:** 到达目标网络 2022:2:2:22::/64 的数据包发给下一跳地址 2012:1:1:11::2。

### (2) 检查静态路由

```
r1#show ipv6 route static
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

S 2022:2:2:22::/64 [1/0]

via 2012:1:1:11::2

r1#

**说明：**从结果中看出，手工配置的递归静态路由已生效。

### (3) 测试连通性

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/48/140 ms

r1#

**说明：**由于正确配置静态路由，R1 到 R2 的 loopback 接口的网段通信正常。

### 3.在 R2 上配置完全静态路由

#### (1) 配置完全静态路由

r2(config)#ipv6 route 2011:1:1:11::/64 f0/0 2012:1:1:11::1

**说明：**到达目标网络 2011:1:1:11::/64 的数据包从接口 F0/0 发出去，并且交给下一跳地址 2012:1:1:11::1。

## (2) 检查静态路由

```
r2#show ipv6 route static
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
S 2011:1:1:11::/64 [1/0]
```

```
via 2012:1:1:11::1, FastEthernet0/0
```

```
r2#
```

**说明：**从结果中看出，手工配置的完全静态路由已生效。

## (3) 测试连通性

```
r2#ping 2011:1:1:11::1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:

!!!!

\*Mar 1 00:36:50.387: %CDP-4-DUPLEX\_MISMATCH: duplex mismatch discovered on FastEthernet0/0 (not full duplex), with Router FastEthernet0/2 (full duplex).!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/92/156 ms

r2#

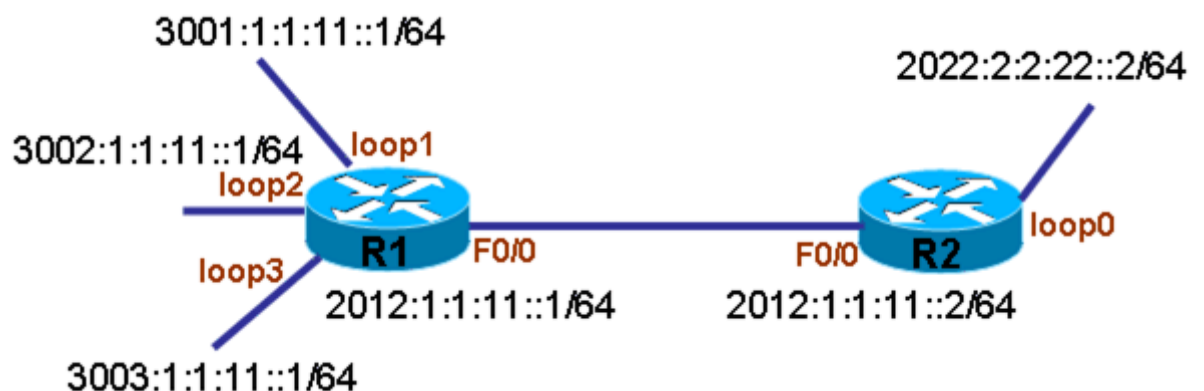
**说明：**由于正确配置静态路由，R2 到 R1 的 loopback 接口的网段通信正常。

## IPv6 RIP (RIPng)

IPv6 的 RIP，所有路由规则与 IPv4 RIPv2 基本相同，不同之处是 IPv4 RIPv2 使用 UDP 端口 520，而 RIPng 使用 UDP 端口 521，IPv4 RIPv2 数据包更新使用地址 224.0.0.9，而 RIPng 使用更新地址为 FF02::9。

在配置 RIPng 时，方法不同于 IPv4 RIP，RIPng 是采用先配置进程，然后需要让哪些接口运行在 RIPng 下，就必须到相应的接口下明确指定，并不像 IPv4 RIP 那样在进程下通过 network 来发布。

### 配置 RIPng



### 1.初始配置

#### (1) R1 初始配置：

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 address 3001:1:1:11::1/64
```

```
r1(config)#int loopback 2
```

```
r1(config-if)#ipv6 address 3002:1:1:11::1/64
```

```
r1(config)#int loopback 3
```

```
r1(config-if)#ipv6 address 3003:1:1:11::1/64
```

## **(2) R2 初始配置:**

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

## **2.启动 RIPng 进程**

**说明:** Cisco IOS 最多同时支持 4 个 RIPng 进程，不同进程使用不同名字来区分，并

且进程名为本地有效。

**(1) 在 R1 上启动 RIPng 进程**

```
r1(config)#ipv6 router rip ccie
```

```
r1(config-rtr)#exit
```

**(2) 在 R2 上启动 RIPng 进程**

```
r2(config)#ipv6 router rip ccie
```

```
r2(config-rtr)#exi
```

**3.配置 RIPng 接口**

**(1) 将 R1 上的接口放进 RIPng 进程**

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 rip ccie enable
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 rip ccie enable
```

**(2) 将 R2 上的接口放进 RIPng 进程**

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 rip ccie enable
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 rip ccie enable
```

**4.查看 RIPng 路由**

**(1) 查看 R1 的 RIPng 路由**



```
r1#show ipv6 route rip
```

IPv6 Routing Table - 11 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
R 2022:2:2:22::/64 [120/2]
```

```
via FE80::C200:DFF:FEC4:0, FastEthernet0/0
```

```
r1#
```

**说明：**由于 RIPng 配置正确，成功收到对方路由条目，并且可以看出，动态路由学习到的 IPv6 路由条目，下一跳地址均为对端的链路本地地址。

## （2）查看 R2 的 RIPng 路由

```
r2#show ipv6 route rip
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
R 3001:1:1:11::/64 [120/2]
```

```
via FE80::C200:BFF:FE48:0, FastEthernet0/0
```

```
r2#
```

**说明：**由于 RIPng 配置正确，成功收到对方路由条目。

## 5.测试连通性

**说明：**因为动态路由学习到的 IPv6 路由条目，下一跳地址均为对端的链路本地地址，所以如果对端的链路本地地址不通，那么到对端 IPv6 网络也不会通。

### (1)测试 R1 到对端链路本地地址的连通性

```
r1#ping FE80::C200:DFF:FEC4:0
```

```
Output Interface: FastEthernet0/0
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::C200:DFF:FEC4:0, timeout is 2 seconds:
```

```
Packet sent with a source address of FE80::C200:DFF:FEC4:0
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/70/184 ms
```

```
r1#
```

**说明：**到对端链路本地地址的通信正常。

### (2) 测试 R1 到对端 IPv6 网络的连通性

```
r1#ping 2022:2:2:22::2
```

```
Type escape sequence to abort.
```

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/75/240 ms

r1

**说明：**由于到对端链路本地地址的通信正常，所以到对端 IPv6 网络的通信也正常。

### (3) 测试 R2 到对端 IPv6 网络的连通性

r2#ping 3001:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3001:1:1:11::1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/84/248 ms

r2#

**说明：**到对端 IPv6 网络的通信也正常。

## 6.重分布 IPv6 网段

**说明：**将 R1 上的剩余网段重分布进 RIPng

### (1) 在 R1 上配置重分布剩余网段进 RIPng

r1(config)#route-map con permit 10

r1(config-route-map)#match interface loopback 2

r1(config-route-map)#exit

r1(config)#route-map con permit 20

r1(config-route-map)#match interface loopback 3

```
r1(config-route-map)#exit
```

```
r1(config)#ipv6 router rip ccie
```

```
r1(config-rtr)#redistribute connected route-map con
```

```
r1(config-rtr)#
```

(2) 在 R2 上查看重分布进 RIPng 的剩余网段

```
r2#show ipv6 route rip
```

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
R 3001:1:1:11::/64 [120/2]
```

```
via FE80::C200:BFF:FE48:0, FastEthernet0/0
```

```
R 3002:1:1:11::/64 [120/2]
```

```
via FE80::C200:BFF:FE48:0, FastEthernet0/0
```

```
R 3003:1:1:11::/64 [120/2]
```

```
via FE80::C200:BFF:FE48:0, FastEthernet0/0
```

```
r2#
```

**说明：**可以看到，R1 上的剩余网段成功被重分布进 RIPng。

## 7.过滤 IPv6 路由

**说明：**在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

### (1) 配置只留 3002:1:1:11::/64 网段

```
r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64
```

```
r2(config)#ipv6 router rip ccie
```

```
r2(config-rtr)#distribute-list prefix-list abc in f0/0
```

**注：**ipv6 的 prefix-list 同样支持 ge , le 等关键字来匹配范围。

### (2) 查看过滤后的路由表情况

```
r2#show ipv6 route rip
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
R 3002:1:1:11::/64 [120/2]
```

```
via FE80::C200:BFF:FE48:0, FastEthernet0/0
```

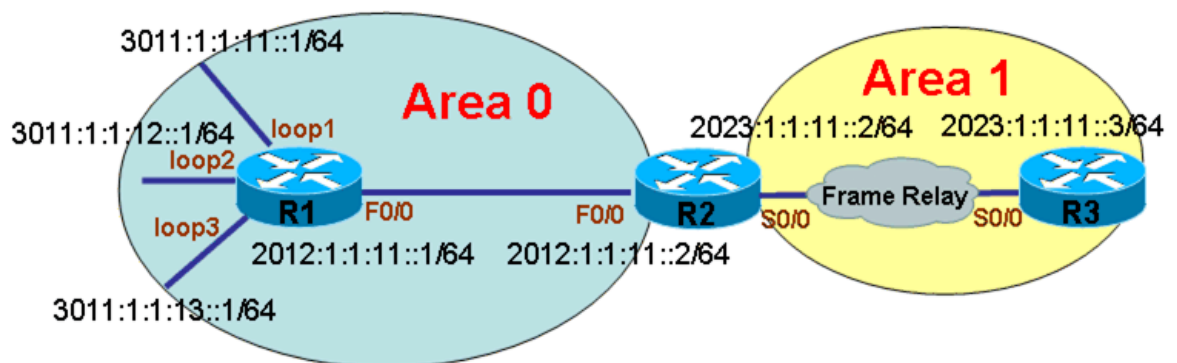
```
r2#
```

**说明：**路由表中只剩想要的网段，说明过滤成功。

## IPv6 OSPF (OSPFv3)

OSPFv3 与 OSPFv2 (IPv4 OSPF) 的原理都是相同的, OSPFv3 选举 Router-ID 的规则与 OSPFv2 相同, OSPFv3 也是选择路由器上的 IPv4 地址作为 Router-ID, 如果设备上没有配置 IPv4 地址, 那么必须手工指定 Router-ID。在配置 OSPFv3 时, 先配置进程, 然后需要让哪些接口运行在 OSPFv3 下, 就必须到相应的接口下明确指定, 并不像 OSPFv2 那样在进程下通过 network 来发布。

### 配置 OSPFv3



#### 1. 初始配置

##### (1) R1 初始配置:

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#interface f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 address 3011:1:1:11::1/64
```

```
r1(config)#int loopback 2
```

```
r1(config-if)#ipv6 address 3011:1:1:12::1/64
```

```
r1(config)#int loopback 3
```

```
r1(config-if)#ipv6 address 3011:1:1:13::1/64
```

## **(2) R2 初始配置:**

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#interface f0/0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config)#interface s1/0
```

```
r2(config-if)#encapsulation frame-relay
```

```
r2(config-if)#no frame-relay inverse-arp
```

```
r2(config-if)#no arp frame-relay
```

```
r2(config-if)#ipv6 address 2023:1:1:11::2/64
```

```
r2(config-if)#frame-relay map ipv6 2023:1:1:11::3 203 broadcast
```

```
r2(config-if)#
```

## **(3) R3 初始配置:**

```
r3(config)#ipv6 unicast-routing
```

```
r3(config)#interface s1/0
```

```
r3(config-if)#encapsulation frame-relay
```

```
r3(config-if)#no frame-relay inverse-arp
```

```
r3(config-if)#no arp frame-relay
```

```
r3(config-if)#ipv6 address 2023:1:1:11::3/64
```

```
r3(config-if)#frame-relay map ipv6 2023:1:1:11::2 302 broadcast
```

## 2.启动 OSPFv3 进程

### (1) 启动 R1 的 OSPFv3 进程

```
r1(config)#ipv6 router ospf 2
```

```
r1(config-rtr)#router-id 1.1.1.1
```

**说明：** 由于没有配置 IPv4 地址，所以必须手工配置 Router-ID

### (2) 启动 R2 的 OSPFv3 进程

```
r2(config)#ipv6 router ospf 2
```

```
r2(config-rtr)#router-id 2.2.2.2
```

### (3) 启动 R3 的 OSPFv3 进程

```
r3(config)#ipv6 router ospf 2
```

```
r3(config-rtr)#router-id 3.3.3.3
```

## 3.配置 OSPFv3 接口

### (1) 将 R1 上的接口放进 OSPFv3 进程

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 ospf 2 area 0
```



```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 ospf 2 area 0
```

(2) 将 R2 上的接口放进 OSPFv3 进程

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 ospf 2 area 0
```

```
r2(config)#int s1/0
```

```
r2(config-if)#ipv6 ospf 2 area 1
```

(3) 将 R3 上的接口放进 OSPFv3 进程

```
r3(config)#int s1/0
```

```
r3(config-if)#ipv6 ospf 2 area 1
```

#### 4.查看 OSPFv3 邻居

(1) 查看 r1 邻居:

```
r1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/BDR	00:00:39	4	FastEthernet0/0

```
r1#
```

**说明:** R1 与 R2 的 OSPFv3 邻居正常。

(2) 查看 r2 邻居:

```
r2#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/DR	00:00:35	4	FastEthernet0/0

```
r2#
```

**说明：**R2 与 R2 的 OSPFv3 邻居正常，但与 R3 的邻居没有。

(3) (3) 查看 r3 邻居：

```
r3#show ipv6 ospf neighbor
```

```
r3#
```

**说明：**R3 没有 OSPFv3 邻居。

## 5. 解决 OSPFv3 邻居问题

**说明：**由于 R2 与 R3 之间属于 NBMA 非广播网络，所以无法自动建邻居，要解决邻居问题，有两种方法：第一，手工指定邻居，在指定时，只须在一方指定即可，并且 OSPFv3 在手工指定邻居时，需要到接口下指定而不是在进程下指定，并且指定的为对方链路本地地址。第二，将网络类型从非广播网络类型改为允许广播的网络类型，如改为 Point-to-point 类型。

(1) 查看 R3 连 R2 接口的链路本地地址

```
r3#show ipv6 interface brief s1/0
```

```
Serial1/0          [up/up]
```

```
FE80::C200:DFF:FEAC:0
```

```
2023:1:1:11::3
```

```
r3#
```

(2) 在 R2 上指定 R3 为邻居，在接口下指定对方的链路本地地址

```
r2(config)#int s1/0
```

```
r2(config-if)#ipv6 ospf neighbor FE80::C200:DFF:FEAC:0
```

```
r2(config-if)#
```

### (3) 测试 R2 到 R3 接口链路本地地址的连通性

```
r2#ping FE80::C200:DFF:FEAC:0
```

Output Interface: Serial1/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FE80::C200:DFF:FEAC:0, timeout is 2 seconds:

Packet sent with a source address of FE80::C200:BFF:FE94:0

.....

Success rate is 0 percent (0/5)

```
r2#
```

**说明：**由于指定邻居时，指定为对方接口的链路本地地址，所以双方接口的链路本地地址不通，邻居将仍然不能建立。

### (4) 解决帧中继网络下双方接口的链路本地地址的 PVC 映射

注：必须互相映射

R2:

```
r2(config)#int s1/0
```

```
r2(config-if)#frame map ipv6 FE80::C200:DFF:FEAC:0 203 broadcast
```

R3:

```
R3(config)#int s1/0
```

```
R3config-if)#frame map ipv6 FE80::C200:BFF:FE94:0 302 broadcast
```

## (5)查看邻居

```
r3#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/BDR	00:01:42	6	Serial1/0

```
r3#
```

**说明：**由于已经手工指定邻居，并且也映射了双方的链路本地地址，所以邻居成功建立。

## 6.查看 OSPFv3 路由

### (1) 在 R1 上查看 OSPFv3 路由

```
r1#sh ipv6 route ospf
```

IPv6 Routing Table - 11 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
OI 2023:1:1:11::/64 [110/74]
```

```
via FE80::C200:BFF:FE94:0, FastEthernet0/0
```

```
r1#
```

**说明：**由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

## (2) 在 R2 上查看 OSPFv3 路由

```
r2#show ipv6 route ospf
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
O  3011:1:1:11::1/128 [110/10]
```

```
via FE80::C200:AFF:FE28:0, FastEthernet0/0
```

```
r2#
```

**说明：**由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

## (3) 在 R3 上查看 OSPFv3 路由

```
r3#show ipv6 route ospf
```

IPv6 Routing Table - 6 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 2012:1:1:11::/64 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

OI 3011:1:1:11::1/128 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

**说明：**由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

## 7.解决 OSPFv3 路由掩码问题

**说明：**由于学习到的路由中，属于 loopback 接口的网段原本为 64 位，而学习到的为 128 位，为主机路由，所以应让路由掩码与原来的掩码一致，需要将网络类型改为 Point-to-point 类型。

**(1) 在 R1 改 loopback 接口的网络类型改为 Point-to-point**

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 ospf network point-to-point
```

```
r1(config-if)#
```

**(2) 查看改后的路由情况**

```
r2#show ipv6 route ospf
```

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 3011:1:1:11::/64 [110/11]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

r2#

**说明：**已经成功变成原来的掩码位数。

## 8.重分布 IPv6 网段

**说明：**将 R1 上的剩余网段重分布进 OSPFv3

**(1) 在 R1 上配置重分布剩余网段进 OSPFv3**

```
r1(config)#route-map con permit 10
```

```
r1(config-route-map)#match interface loopback 2
```

```
r1(config-route-map)#exit
```

```
r1(config)#route-map con permit 20
```

```
r1(config-route-map)#match interface loopback 3
```

```
r1(config-route-map)#exit
```

```
r1(config)#ipv6 router ospf 2
```

```
r1(config-rtr)#redistribute connected route-map con
```

**(2) 在 R2 上查看重分布进 OSPFv3 的剩余网段**

```
r2#show ipv6 route ospf
```

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 3011:1:1:11::/64 [110/11]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

r2#

**说明：**可以看到，R1 上的剩余网段成功被重分布进 OSPFv3。

**(3) 在 R3 上查看重分布进 OSPFv3 的剩余网段**

r3#show ipv6 route ospf

IPv6 Routing Table - 8 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2



ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 2012:1:1:11::/64 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

OI 3011:1:1:11::/64 [110/75]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

**说明：**可以看到，R1 上的剩余网段成功被重分布进 OSPFv3。

## 9.过滤 IPv6 路由

**说明：**在 R3 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

### （1）配置只留 3011 打头的网段

```
r3(config)#ipv6 prefix-list abc permit 3011::/16 ge 64 le 64
```

```
r3(config)#ipv6 router ospf 2
```

```
r3(config-rtr)#distribute-list prefix-list abc in s1/0
```

### （2）查看过滤后的路由表情况

```
r3#show ipv6 route ospf
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 3011:1:1:11::/64 [110/75]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

**说明：**路由表中只剩 3011 打头的网段了，说明过滤成功。

## 10. 汇总 OSPFv3 外部路由

**说明：**对从外部重分布进 OSPFv3 的路由进行汇总，OSPF 内的路由汇总，命令格式基本同 IPv4，需要注意的是，汇总必须在重分布的路由器上配置，即必须在 ASBR 上配置。

### （1）在 ASBR（R1）上配置外部路由的汇总

**说明：**将 3011:1:1:11::/64，3011:1:1:12::/64，3011:1:1:13::/64 三条路由汇总成 3011:1:1::/48

r1(config)#ipv6 router ospf 2

r1(config-rtr)#summary-prefix 3011:1:1::/48

r1(config-rtr)#

(2)在 R2 上查看汇总后的路由表情况

```
r2#show ipv6 route ospf
```

IPv6 Routing Table - 8 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
OE2  3011:1:1::/48 [110/20]
```

```
via FE80::C200:AFF:FE28:0, FastEthernet0/0
```

```
O   3011:1:1:11::/64 [110/11]
```

```
via FE80::C200:AFF:FE28:0, FastEthernet0/0
```

```
r2#
```

**说明：**可以看到，汇总成功。

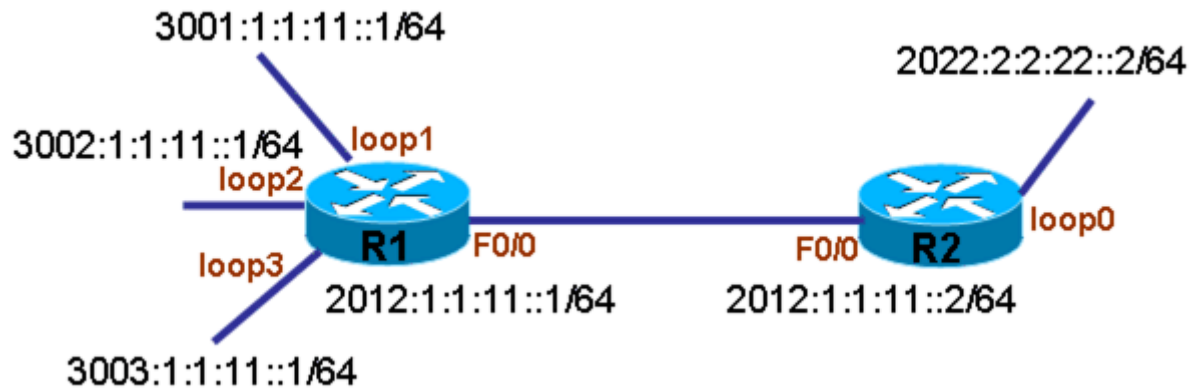
## IPv6 EIGRP (EIGRP v6)

EIGRP v6 与 IPv4 EIGRP 的原理都是相同的，但是 EIGRP v6 必须有 router-id 才能运行，所以在 EIGRP v6 不能获得 router-id 时，请手工配置 router-id；更多的是 EIGRP v6 进程有个 shutdown 的特性，要用 no shutdown 开启进程；在配置 EIGRP v6 时，先配置进程，然后需要让哪些接口运行在 EIGRP v6 下，就必须到相应的接口下明确指定，并不像 IPv4 EIGRP 那样通过 network 来发布。

EIGRP hello 时间默认是 5 秒一个，在低链路是 60 秒一个，比如 NBMA，或者所

有低于或等于 T1 的链路（1.544M）。Hold time 是 hello 的三倍。

### 配置 EIGRP v6



#### 1.初始配置

(1) R1 初始配置:

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 address 3001:1:1:11::1/64
```

```
r1(config)#int loopback 2
```

```
r1(config-if)#ipv6 address 3002:1:1:11::1/64
```

```
r1(config)#int loopback 3
```

```
r1(config-if)#ipv6 address 3003:1:1:11::1/64
```

## (2) R2 初始配置:

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

## 2.配置 EIGRP v6 进程

### (1) 在 R1 上启动 EIGRP v6 进程

```
r1(config)#ipv6 router eigrp 10
```

```
r1(config-rtr)#router-id 1.1.1.1
```

### (2) 在 R2 上启动 EIGRP v6 进程

```
r2(config)#ipv6 router eigrp 10
```

```
r2(config-rtr)#router-id 2.2.2.2
```

## 3.配置 EIGRP v6 接口

### (1) 将 R1 上的接口放进 EIGRP v6 进程

```
r1(config)#interface f0/0
```

```
r1(config-if)#ipv6 eigrp 10
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 eigrp 10
```

### (2) 将 R2 上的接口放进 EIGRP v6 进程

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 eigrp 10
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 eigrp 10
```

### (3) 查看 EIGRP v6 邻居状态

```
r1#show ipv6 eigrp neighbors
```

```
IPv6-EIGRP neighbors for process 10
```

```
% EIGRP 10 is in SHUTDOWN
```

```
r1#
```

**说明：**从结果中看出，EIGRP v 进程默认是 shutdown 的，必须手工开启。

### (4) 开启 EIGRP v6 进程

```
r1(config)#ipv6 router eigrp 10
```

```
r1(config-rtr)#no shutdown
```

### (5) 查看邻居

```
r1#show ipv6 eigrp neighbors
```

IPv6-EIGRP neighbors for process 10

H	Address	Interface	Hold Uptime	SRTT	RTO	Q	Seq
		(sec)	(ms)	Cnt	Num		
0	Link-local address: FE80::C200:AFF:FE50:0	Fa0/0	11 00:00:36	192	1152	0	2

**说明：** 开启 EIGRP v6 进程后，邻居正常建立。

#### 4.查看 EIGRP v6 路由

##### (1) 查看 R1 的 EIGRP v6 路由

```
r1#show ipv6 route eigrp
```

IPv6 Routing Table - 11 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
D 2022:2:2:22::/64 [90/409600]
```

```
via FE80::C200:AFF:FE50:0, FastEthernet0/0
```

```
r1#
```

**说明：** 由于 EIGRP v6 配置正确，成功收到对方路由条目。

##### (2) 查看 R2 的 EIGRP v6 路由

```
r2#sh ipv6 route eigrp
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

D 3001:1:1:11::/64 [90/409600]

via FE80::C200:9FF:FE54:0, FastEthernet0/0

r2#

**说明：**由于 EIGRP v6 配置正确，成功收到对方路由条目。

## 5.重分布 IPv6 网段

**说明：**将 R1 上的剩余网段重分布进 EIGRP v6

**(1) 在 R1 上配置重分布剩余网段进 EIGRP v6**

```
r1(config)#route-map con permit 10
```

```
r1(config-route-map)#match interface loopback 2
```

```
r1(config-route-map)#exit
```

```
r1(config)#route-map con permit 20
```

```
r1(config-route-map)#match interface loopback 3
```

```
r1(config)#ipv6 router eigrp 10
```

```
r1(config-rtr)#redistribute connected route-map con
```



```
r1(config-rtr)#exit
```

(2) 在 R2 上查看重分布进 EIGRP v6 的剩余网段

```
r2#sh ipv6 route eigrp
```

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
D 3001:1:1:11::/64 [90/409600]
```

```
via FE80::C200:9FF:FE54:0, FastEthernet0/0
```

```
EX 3002:1:1:11::/64 [170/409600]
```

```
via FE80::C200:9FF:FE54:0, FastEthernet0/0
```

```
EX 3003:1:1:11::/64 [170/409600]
```

```
via FE80::C200:9FF:FE54:0, FastEthernet0/0
```

```
r2#
```

**说明：**可以看到，R1 上的剩余网段成功被重分布进 EIGRP v6。

## 6. 过滤 IPv6 路由

**说明：**在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

(1) 配置只留 3002:1:1:11::/64 网段

```
r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64
```

```
r2(config)#ipv6 router eigrp 10
```

```
r2(config-rtr)#distribute-list prefix-list abc in f0/0
```

```
r2(config-rtr)#
```

## (2) 查看过滤后的路由表情况

```
r2#sh ipv6 route eigrp
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
EX 3002:1:1:11::/64 [170/409600]
```

```
via FE80::C200:9FF:FE54:0, FastEthernet0/0
```

```
r2#
```

**说明：**路由表中只剩想要的网段，说明过滤成功。

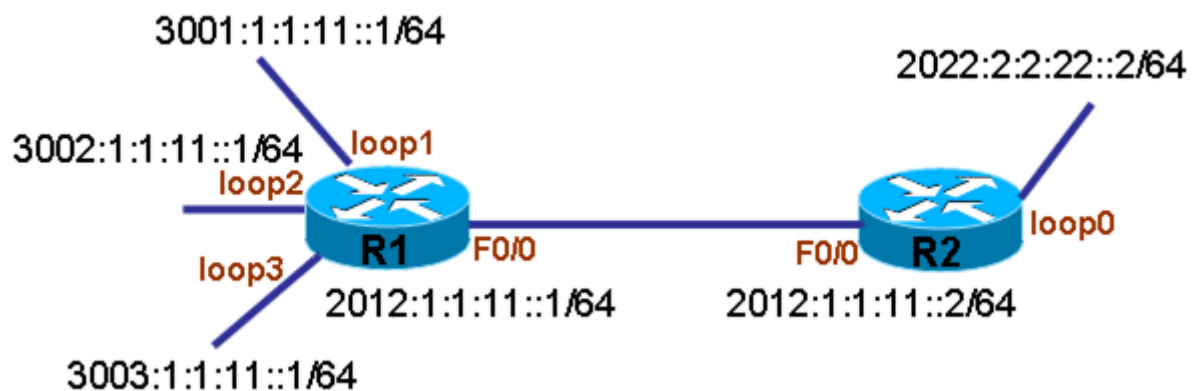
## IPv6 BGP

普通情况下配置的 BGP，是用来传递 IPv4 路由的，所传递的信息是 IPv4 的协议，如果要让 BGP 传递其它路由或协议，这就需要将 BGP 扩展为支持更多协议的 BGP，如扩展 BGP 支持 IPv6 协议，支持 vpnv4，这样的支持多协议的 BGP，称为 Multiprotocol

BGP，即 MP-BGP，

要配置 MP-BGP，就需要为除 IPv4 之外的协议单独创建 address-family，但是建立 BGP 邻居和正常情况下一样，当邻居建立之后，还得到 address-family 下活动，这是 MP-BGP 的特性，而需要发布的网段，也需要到 address-family 下发布。传递单播 IPv6 的 address-family 应该是 address-family ipv6 unicast，但关键字 unicast 如果省略，默认就是 address-family ipv6 unicast。下面根据以上特征，来配置 MP-BGP 传递 IPv6 路由。

### 配置 IPv6 MP-BGP



#### 1.初始配置

##### (1) R1 初始配置:

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config)#int loopback 1
```

```
r1(config-if)#ipv6 address 3001:1:1:11::1/64
```

```
r1(config)#int loopback 2
```

```
r1(config-if)#ipv6 address 3002:1:1:11::1/64
```

```
r1(config)#int loopback 3
```

```
r1(config-if)#ipv6 address 3003:1:1:11::1/64
```

## (2) R2 初始配置:

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#int f0/0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

## 2.配置 MP-BGP 中的 IPv6 邻居

**说明:** 所有邻居正常配置，但需要到 IPv6 的 address-family 下激活邻居。

### (1) 在 R1 上配置 BGP 邻居

```
r1(config)#router bgp 100
```

```
r1(config-router)#bgp router-id 1.1.1.1
```

```
r1(config-router)#neighbor 2012:1:1:11::2 remote-as 100
```

```
r1(config-router)#address-family ipv6
```

```
r1(config-router-af)#neighbor 2012:1:1:11::2 activate
```

```
r1(config-router-af)#exit
```

## (2) 在 R2 上配置 BGP 邻居

```
r2(config)#router bgp 100
```

```
r2(config-router)#bgp router-id 2.2.2.2
```

```
r2(config-router)#neighbor 2012:1:1:11::1 remote-as 100
```

```
r2(config-router)#address-family ipv6
```

```
r2(config-router-af)#neighbor 2012:1:1:11::1 activate
```

```
r2(config-router-af)#exit
```

## 3. 查看 IPv6 BGP 邻居

### (1) 在 R1 上查看 IPv6 BGP 邻居

```
r1#show bgp sum
```

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2012:1:1:11::2	4	100	5	4	1	0	0	00:01:35	0

```
r1#
```

**说明：**由于配置正确，所以已正常建立 IPv6 BGP 邻居命令。命令 `show bgp sum` 为隐藏命令。

(2) 在 R2 上查看 IPv6 BGP 邻居

```
r2#show bgp sum
```

BGP router identifier 2.2.2.2, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2012:1:1:11::1	4	100	5	6	1	0	0	00:02:02	0

```
r2#
```

**说明：**由于配置正确，所以已正常建立 IPv6 BGP 邻居命令。

#### 4.发布 IPv6 路由进 IPv6 BGP

(1)在 R1 上发布路由进 IPv6 BGP

```
r1(config)#router bgp 100
```

```
r1(config-router)#address-family ipv6
```

```
r1(config-router-af)#network 3001:1:1:11::/64
```

(2)在 R2 上发布路由进 IPv6 BGP

```
r2(config)#router bgp 100
```

```
r2(config-router)#address-family ipv6
```

```
r2(config-router-af)#network 2022:2:2:22::/64
```

(3)在 R1 上查看 IPv6 BGP 路由

```
r1#show bgp all
```

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i2022:2:2:22::/64	2012:1:1:11::2	0	100	0	i
*> 3001:1:1:11::/64 ::		0	32768		i

r1#

**说明：**已成功学习到对方邻居发来的 IPv6 路由。

#### (4)在 R2 上查看 IPv6 BGP 路由

r2#show bgp all

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2022:2:2:22::/64 ::		0	32768		i
*>i3001:1:1:11::/64	2012:1:1:11::1	0	100	0	i

r2#

**说明：**已成功学习到对方邻居发来的 IPv6 路由。

### (5) 测试网络连通性

```
r1#ping 2022:2:2:22::2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/96/208 ms

```
r1#
```

```
r2#ping 3001:1:1:11::1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3001:1:1:11::1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 24/88/200 ms

```
r2#
```

**说明：**由于双方路由学习正常，所以网络连通性正常。

### 5.重分布 IPv6 网段

**说明：**将 R1 上的剩余网段重分布进 IPv6 BGP

#### (1) 在 R1 上配置重分布剩余网段进 IPv6 BGP

```
r1(config)#route-map con permit 10
```

```
r1(config-route-map)#match interface loopback 2
```



```
r1(config-route-map)#exit
```

```
r1(config)#route-map con permit 20
```

```
r1(config-route-map)#match interface loopback 3
```

```
r1(config-route-map)#exit
```

```
r1(config)#router bgp 100
```

```
r1(config-router)#address-family ipv6
```

```
r1(config-router-af)#redistribute connected route-map con
```

(2) 在 R2 上查看重分布进 IPv6 BGP 的剩余网段

```
r2#show bgp all
```

For address family: IPv6 Unicast

BGP table version is 11, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2022:2:2:22::/64 ::		0	32768	i	
*>i3001:1:1:11::/64 2012:1:1:11::1		0	100	0	i
*>i3002:1:1:11::/64 2012:1:1:11::1		0	100	0	?
*>i3003:1:1:11::/64 2012:1:1:11::1		0	100	0	?

r2#

**说明：**可以看到，R1 上的剩余网段成功被重分布进 RIPv6。

## 6. 过滤 IPv6 路由

**说明：**在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 对指定邻居进行过滤

### (1) 配置只留 3002:1:1:11::/64 网段

```
r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64
```

```
r2(config)#router bgp 100
```

```
r2(config-router)#address-family ipv6
```

```
r2(config-router-af)#neighbor 2012:1:1:11::1 prefix-list abc in
```

### (2) 查看过滤后的路由表情况

```
r2#clear bgp ipv6 unicast *
```

```
r2#sh bgp all
```

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2022:2:2:22::/64 ::		0		32768	i

```
*>i3002:1:1:11::/64 2012:1:1:11::1 0 100 0 ?
```

```
r2#
```

**说明：**路由表中只剩想要的网段，说明过滤成功。

## 7.使用链路本地地址建立 IPv6 BGP 邻居

**说明：**正常情况下，IPv6 BGP 使用全局地址建立邻居，也可以配置使用链路本地地址建立邻居。

### （1）在 R1 上配置 IPv6 BGP 用链路本地地址建立邻居

```
r1(config)#router bgp 100
```

```
r1(config-router)#neighbor FE80::C200:DFF:FEC8:0 remote-as 100
```

```
r1(config-router)#neighbor FE80::C200:DFF:FEC8:0 update-source f0/0
```

```
r1(config-router)#address-family ipv6
```

```
r1(config-router-af)#neighbor FE80::C200:DFF:FEC8:0 activate
```

```
r1(config-router-af)#
```

### （2）在 R2 上配置 IPv6 BGP 用链路本地地址建立邻居

```
r2(config)#router bgp 100
```

```
r2(config-router)#neighbor FE80::C200:8FF:FE10:0 remote-as 100
```

```
r2(config-router)#neighbor FE80::C200:8FF:FE10:0 update-source f0/0
```

```
r2(config-router)#address-family ipv6
```

```
r2(config-router-af)#neighbor FE80::C200:8FF:FE10:0 activate
```

```
r2(config-router-af)#
```

### （3）查看邻居建立情况

```
r1#show bgp sum
```

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
----------	---	----	---------	---------	--------	-----	------	---------	--------------

FE80::C200:DFF:FEC8:0

4	100	6	7	1	0	0	00:01:30	0
---	-----	---	---	---	---	---	----------	---

r1#

r2#show bgp sum

BGP router identifier 2.2.2.2, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
----------	---	----	---------	---------	--------	-----	------	---------	--------------

FE80::C200:8FF:FE10:0

4	100	8	7	1	0	0	00:02:20	0
---	-----	---	---	---	---	---	----------	---

r2#

**说明：**从结果中看出，双方 IPv6 BGP 已成功使用链路本地地址建立邻居

#### (4) 查看路由学习情况

r1#sh bgp all

For address family: IPv6 Unicast

BGP table version is 14, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

\*>i2022:2:2:22::/64 FE80::C200:DFF:FEC8:0

0	100	0	i
---	-----	---	---

\*> 3001:1:1:11::/64 :: 0 32768 i

\*> 3002:1:1:11::/64 :: 0 32768 ?

\*> 3003:1:1:11::/64 :: 0 32768 ?

r1#

r2#show bgp all

For address family: IPv6 Unicast

BGP table version is 6, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

\*> 2022:2:2:22::/64 :: 0 32768 i

\*>i3001:1:1:11::/64 FE80::C200:8FF:FE10:0

```
0 100 0 i
```

```
*>i3002:1:1:11::/64 FE80::C200:8FF:FE10:0
```

```
0 100 0 ?
```

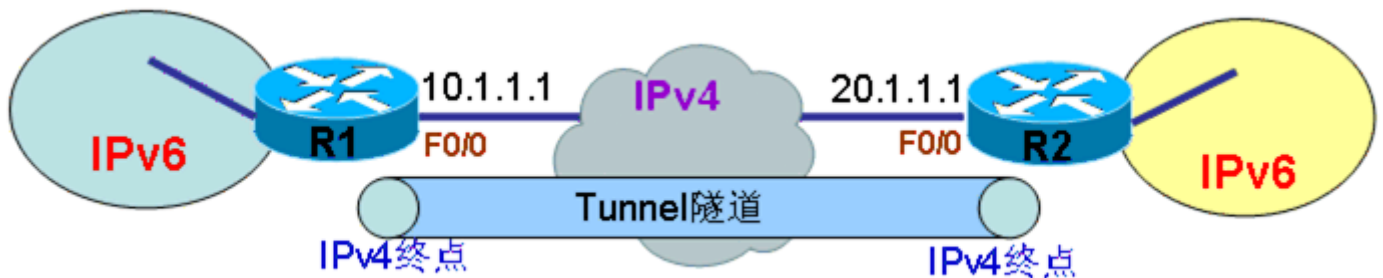
```
*>i3003:1:1:11::/64 FE80::C200:8FF:FE10:0
```

```
0 100 0 ?
```

```
r2#
```

**说明：**从结果中看出，双方 IPv6 BGP 已成功学习到相互的 IPv6 路由条目。

## IPv6 隧道



如上图所示，当两个 IPv6 网络需要通信时，如果中间需要穿越 IPv4 网络，而由于 IPv4 网络中只能识别 IPv4 包头，并不能为 IPv6 数据提供正确的路径传输，这时就需要在 IPv4 网络中为 IPv6 创建一条隧道，来提供 IPv6 在 IPv4 中的传递，这样的隧道，就是把 IPv6 的数据全部封装在 IPv4 中，将 IPv4 当作链路层来传递的隧道形式，称为覆盖型隧道（Overlay Tunnels）。由于隧道是建立在 IPv4 基础上的，隧道又必须有起点和终点来明确隧道的路径，所以覆盖型隧道的起点和终点最好是使用 IPv4 地址，有时必须是 IPv4 地址，并且隧道在传输 IPv6 数据时，也应该在隧道的两端添加 IPv6 地址，来完成两端 IPv6 网络的通信。隧道的起点和终点必须同时支持 IPv4 和 IPv6。

当前在 IOS 中支持的覆盖型隧道共有以下几中：

#### **Manual**

点对点，只传递 IPv6 数据包。

模式为： ipv6ip

#### **Generic routing encapsulation (GRE)**

点对点，可以传递多种协议。

模式为： gre ip

#### **IPv4-compatible**

点到多点的，思科不建议使用。

模式为： ipv6ip auto-tunnel

#### **6to4**

点到多点的，使用地址为 2002::/16。

模式为： ipv6ip 6to4

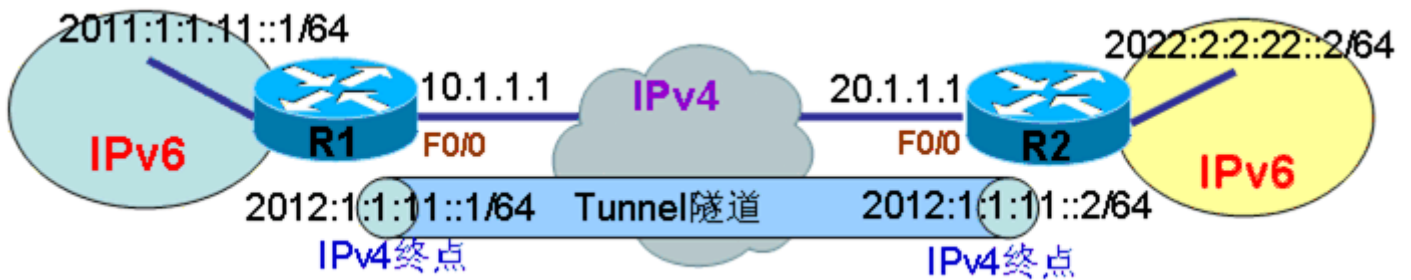
#### **Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)**

是点到多点的。

模式为： ipv6ip isatap

以上隧道中，所有隧道的源均为 IPv4 地址，但是只有点对点隧道的终点为 IPv4 地址，

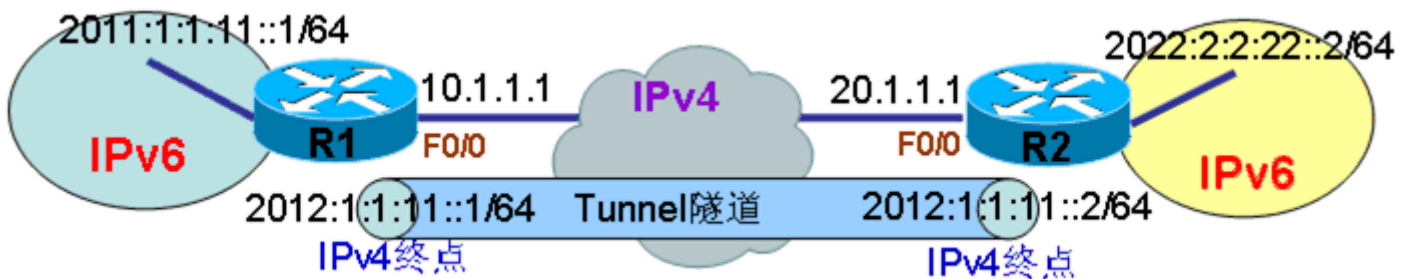
其它都不需要。更多的是，点对点隧道必须要有 IPv6 地址，点对点隧道如下图：



**注：**CCIE 考试中，IPv6 隧道的考点为 Manual 类型的隧道。

#### 配置 IPv6 隧道：

**说明：**原来 R1 上的 IPv6 网络无法与 R2 上的 IPv6 网络通信，通过配置 IPv6 隧道之后，在隧道与相应 IPv6 接口上启用 IPv6 路由协议，如 OSPFv3，使得两端 IPv6 网络可以通信。



#### 1.初始配置

(1) R1 初始配置：

```
r1(config)#int f0/0
```

```
r1(config-if)#ip address 10.1.1.1 255.255.255.0
```

```
r1(config-if)#exi
```



```
r1(config)#ip route 0.0.0.0 0.0.0.0 f0/0
```

```
r1(config)#
```

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int loopback 0
```

```
r1(config-if)#ipv6 address 2011:1:1:11::1/64
```

## **(2) R2 初始配置:**

```
r2(config)#int f0/1
```

```
r2(config-if)#ip add 20.1.1.1 255.255.255.0
```

```
r2(config-if)#exit
```

```
r2(config)#ip route 0.0.0.0 0.0.0.0 f0/0
```

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

## **测试 IPv4 连通性:**

```
r1#ping 20.1.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/108/196 ms

r1#

**说明：**R1 与 R2 之间的 IPv4 连通性正常。

## 2.配置 Manual 类型的 IPv6 隧道

### （1）在 R1 上配置 IPv6 隧道

**说明：**配置的 IPv6 隧道的类型为 Manual 类型，即模式为 ipv6ip

```
r1(config)#int tunnel 0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config-if)#tunnel source f0/0
```

```
r1(config-if)#tunnel destination 20.1.1.1
```

```
r1(config-if)#tunnel mode ipv6ip
```

### （2）在 R2 上配置 IPv6 隧道

```
r2(config)#int tunnel 0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config-if)#tunnel source f0/0
```

```
r2(config-if)#tunnel destination 10.1.1.1
```

```
r2(config-if)#tunnel mode ipv6ip
```

### (3)查看两端隧道情况

```
r1#sh ipv6 interface brief tunnel 0
```

```
Tunnel0          [up/up]
```

```
FE80::A01:101
```

```
2012:1:1:11::1
```

```
r1#
```

```
r2#show ipv6 interface brief tunnel 0
```

```
Tunnel0          [up/up]
```

```
FE80::1401:101
```

```
2012:1:1:11::2
```

```
r2#
```

### (4) 测试隧道连通性:

```
r1#ping 2012:1:1:11::2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 2012:1:1:11::2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 212/279/332 ms
```

```
r1#
```

**说明：**隧道通信正常。

### 3.配置 IPv6 路由协议

**说明：**在路由器之间启用 IPv6 路由协议，在隧道上传递两端 IPv6 网络信息。

#### (1)在 R1 上配置 OSPFv3

```
r1(config)#ipv6 router ospf 2
```

```
r1(config-rtr)#router-id 1.1.1.1
```

```
r1(config)#int loopback 0
```

```
r1(config-if)#ipv6 ospf network point-to-point
```

```
r1(config-if)#ipv6 ospf 2 area 0
```

```
r1(config)#int tunnel 0
```

```
r1(config-if)#ipv6 ospf 2 area 0
```

#### (2)在 R2 上配置 OSPFv3

```
r2(config)#ipv6 router ospf 2
```

```
r2(config-rtr)#router-id 2.2.2.2
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 ospf network point-to-point
```

```
r2(config-if)#ipv6 ospf 2 area 0
```

```
r2(config)#int tunnel 0
```

```
r2(config-if)#ipv6 ospf 2 area 0
```

#### 4.查看结果

##### (1) 查看邻居状态

R1:

```
r1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/ -	00:00:31	14	Tunnel0

r1#

R2:

```
r2#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/ -	00:00:38	14	Tunnel0

r2

**说明：**两端 OSPFv3 邻居正常。

##### (2) 查看路由信息

R1:

```
r1#show ipv6 route ospf
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2022:2:2:22::/64 [110/11112]

via FE80::1401:101, Tunnel0

r1#

R2:

r2#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2011:1:1:11::/64 [110/11112]

via FE80::A01:101, Tunnel0

r2#

**说明：**两端通过 OSPFv3 学习到的 IPv6 路由正常。

## 5.测试两端 IPv6 网络通信情况

### (1) 测试 R1 到 R2 端 IPv6 网络的通信情况

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 120/181/268 ms

r1#

**说明：**由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

## （2）测试 R2 到 R1 端 IPv6 网络的通信情况

r2#ping 2011:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 88/164/284 ms

r2#

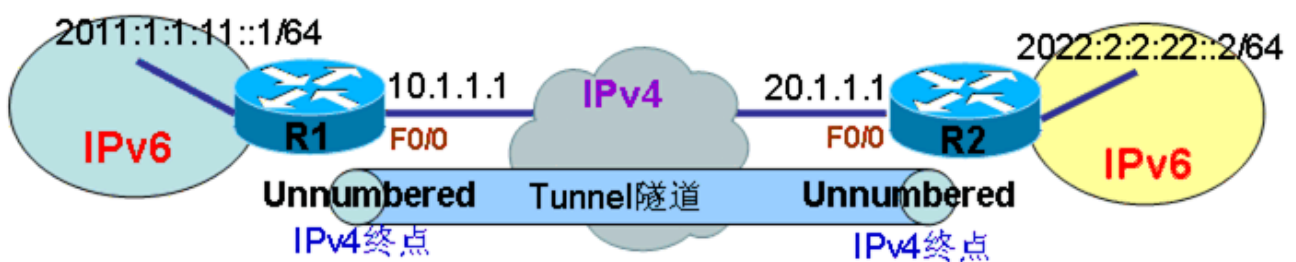
**说明：**由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

## 隧道借用地址

**说明：**在现有 IPv4 网络上创建覆盖型 IPv6 隧道，隧道的起点和终点都使用了 IPv4 地址来定义，然后要使隧道运行正常，使隧道具有路由协议的连接功能，需要赋予隧道两端 IPv6 地址，从而提供 IPv6 的连通性，而隧道两端的 IPv6 地址可以不属于

同一网段，当然属于同一网段是最好的选择。无论隧道两端的 IPv6 地址是否属于同一网段，IPv6 路由协议都是可以正常使用的。如果隧道两端的 IPv6 地址属于同一网段，那么一切正常，隧道两端的地址可以相互 ping 通，路由协议也无须更多操作，而当隧道两端的 IPv6 地址不属于同一网段时，那么两端的地址是无法 ping 通的，但 IPv6 路由协议可以照常使用，这时，路由协议需要将隧道的地址当作额外路由进行重新通告一次。

下面在创建隧道时，将隧道两端的 IPv6 地址改为无编号借用地址（unnumbered），这时两端地址不属于同网段，再使用 IPv6 路由协议连通两端 IPv6 网络。



### 1.初始配置

r1:

```
r1(config)#int f0/0
```

```
r1(config-if)#ip address 10.1.1.1 255.255.255.0
```

```
r1(config-if)#exi
```

```
r1(config)#ip route 0.0.0.0 0.0.0.0 f0/0
```

```
r1(config)#
```

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#int loopback 0
```



```
r1(config-if)#ipv6 address 2011:1:1:11::1/64
```

**R2**

```
r2(config)#int f0/1
```

```
r2(config-if)#ip add 20.1.1.1 255.255.255.0
```

```
r2(config-if)#exit
```

```
r2(config)#ip route 0.0.0.0 0.0.0.0 f0/0
```

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

**测试 IPv4 连通性:**

```
r1#ping 20.1.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/108/196 ms

```
r1#
```

**说明：**R1 与 R2 之间的 IPv4 连通性正常。

## 2.配置 unnumbered 地址的 IPv6 隧道

### (1) 在 R1 上配置 IPv6 隧道

```
r1(config)#int tunnel 0
```

```
r1(config-if)#ipv6 unnumbered loopback 0
```

```
r1(config-if)#tunnel source f0/0
```

```
r1(config-if)#tunnel destination 20.1.1.1
```

```
r1(config-if)#tunnel mode ipv6ip
```

### (2) 在 R2 上配置 IPv6 隧道

```
r2(config)#int tunnel 0
```

```
r2(config-if)#ipv6 unnumbered loopback 0
```

```
r2(config-if)#tunnel source f0/0
```

```
r2(config-if)#tunnel destination 10.1.1.1
```

```
r2(config-if)#tunnel mode ipv6ip
```

### (3)查看两端隧道情况

```
r1#show ipv6 interface brief tunnel 0
```

```
Tunnel10          [up/up]
```

```
FE80::A01:101
```

```
unnumbered (Loopback0)
```

```
r1#
```

```
r2#show ipv6 interface brief tunnel 0
```

```
Tunnel10          [up/up]
```

```
FE80::1401:101
```

```
unnumbered (Loopback0)
```

```
r2#
```

#### (4) 测试隧道连通性:

```
r1#ping 2022:2:2:22::2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:
```

```
.....
```

```
Success rate is 0 percent (0/5)
```

```
r1#
```

**说明:** 由于隧道两端地址不属于同一网段，所以没有对端路由信息，无法 ping 通。

### 3.配置 IPv6 路由协议

#### (1)在 R1 上配置 OSPFv3

```
r1(config)#ipv6 router ospf 10
```

```
r1(config-rtr)#router-id 1.1.1.1
```

```
r1(config)#int loopback 0
```

```
r1(config-if)#ipv6 ospf network point-to-point
```

```
r1(config-if)#ipv6 ospf 10 area 0
```

```
r1(config)#int tunnel 0
```

```
r1(config-if)#ipv6 ospf 10 area 0
```

## **(2)在 R2 上配置 OSPFv3**

```
r2(config)#ipv6 router ospf 10
```

```
r2(config-rtr)#router-id 2.2.2.2
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 ospf network point-to-p
```

```
r2(config-if)#ipv6 ospf 10 area 0
```

```
r2(config)#int tunnel 10
```

```
r2(config-if)#ipv6 ospf 10 area 0
```

## **4.查看结果**

### **(1) 查看邻居状态**

```
r1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/ -	00:00:37 15		Tunnel10

```
r1#
```

```
r2#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/ -	00:00:36 15		Tunnel10

```
r2#
```

**说明：**两端 OSPFv3 邻居正常。

## (2) 查看路由信息

```
R1:
```

```
r1#show ipv6 route ospf
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
O  2022:2:2:22::/64 [110/11112]
```

```
via FE80::1401:101, Tunnel10
```

```
r1#
```

```
R2:
```

```
r2#show ipv6 route ospf
```

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2011:1:1:11::/64 [110/11112]

via FE80::A01:101, Tunnel10

r2#

**说明：**两端通过 OSPFv3 学习到的 IPv6 路由正常。

## 5.测试两端 IPv6 网络通信情况

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 120/181/268 ms

r1#

**说明：**由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

## (2) 测试 R2 到 R1 端 IPv6 网络的通信情况

```
r2#ping 2011:1:1:11::1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/164/284 ms
```

```
r2#
```

**说明：**由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

## IPv6 组播

### 概述

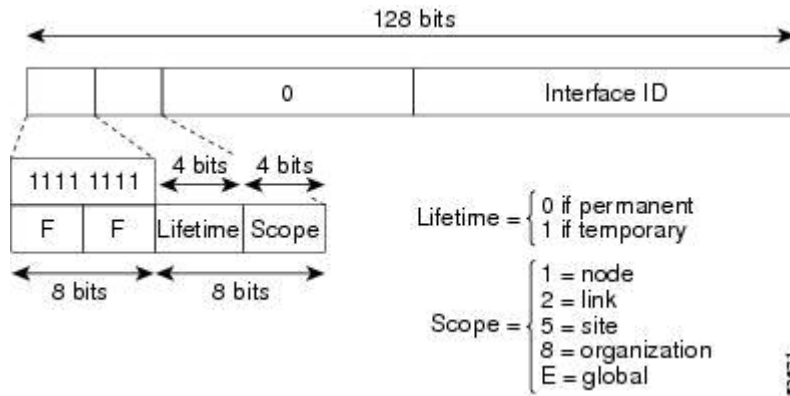
在理解 IPv6 组播之前，必须了解 IPv4 组播，了解 IPv4 PIM，了解 IGMP，这些知识在本篇不再详细讲述，相关详细内容，请参见 IPv4 组播部分。

要启用 IPv6 组播，必须先开 IPv6 单播。

### IPv6 组播地址：

IPv6 组播地址的范围是 FF00::/8 (1111 1111)。

因为一个正常的 IPv6 地址包含 128 位，在 IPv6 组播地址中，第一段共 16 位的格式被拆分成三部分：第一部分共 8 位，全部为 1，即使用 FF 来表示。第二部分共 4 位，表示组播地址的存活期，如果为 0 表示永久，如果为 1 表示临时。第三部分共 4 位，表示组播地址的范围，分为 node, link, site, organization, global 分别表示为 1, 2, 5, 8, E，除了此五种以外，0 和 F 为保留范围，而其它全部称为未分配，建议使用未分配的地址范围。组播地址的表示格式如下图：



在 IPv6 中没有广播地址，只有组播，所以使用组播代替广播。

无论是路由器还是主机，所有 IPv6 接口默认加入 FF02::1 ，

而所有路由器的 IPv6 接口默认加入 FF02::2。

### MLD （Multicast Listener Discovery）

在 IPv6 组播中，MLD 协议与 IPv4 组播中的 IGMP 协议功能相同，是用于发现接收者的协议。

路由器发送 MLD 查询消息来确认接收者，而主机发送 MLD 报告来加入一个组，主机可以在同一时间属于多个组。

MLD 共有两个版本，ver1 和 ver2，

MLD ver 1 是基于 IPv4 IGMP v2

MLD ver 2 是基于 IPv4 IGMP v3

IOS 同时使用两个。

### PIM

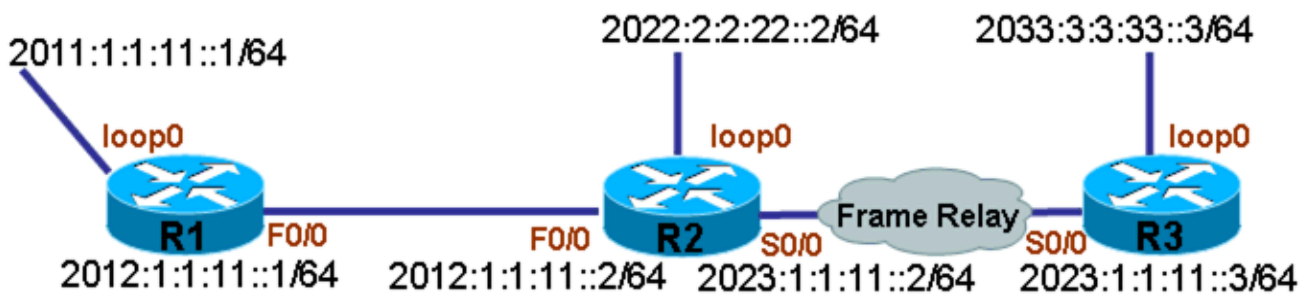
IPv6 PIM 的功能同 IPv4 PIM，而 IPv6 PIM 只使用 SM（稀疏）模式，所以网络中必须存在 RP，而 RP 的位置可以通过静态配置和 BSR 通告两种方法确认。

在配置 IPv6 PIM 时，当开启 IPv6 组播功能后，所有正常启用 IPv6 功能的接口自



动开启 IPv6 PIM，所以 IPv6 PIM 无须手工配置；并且须明白 DR 在组播中的作用，详细内容请参见 IPv4 组播部分。

### 配置 IPv6 组播



#### 1.初始配置

##### (1) R1 初始配置:

```
r1(config)#ipv6 unicast-routing
```

```
r1(config)#ipv6 router ospf 10
```

```
r1(config-rtr)#router-id 1.1.1.1
```

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2012:1:1:11::1/64
```

```
r1(config-if)#ipv6 ospf 10 area 0
```

```
r1(config)#int loopback 0
```

```
r1(config-if)#ipv6 address 2011:1:1:11::1/64
```

```
r1(config-if)#ipv6 ospf network point-to-point
```

```
r1(config-if)#ipv6 ospf 10 area 0
```

**(2) R2 初始配置:**

```
r2(config)#ipv6 unicast-routing
```

```
r2(config)#ipv6 router ospf 10
```

```
r2(config-rtr)#router-id 2.2.2.2
```

```
r2(config)#interface f0/0
```

```
r2(config-if)#ipv6 address 2012:1:1:11::2/64
```

```
r2(config-if)#ipv6 ospf 10 area 0
```

```
r2(config)#int loopback 0
```

```
r2(config-if)#ipv6 address 2022:2:2:22::2/64
```

```
r2(config-if)#ipv6 ospf network point-to-point
```

```
r2(config-if)#ipv6 ospf 10 area 0
```

```
r2(config)#int s0/0
```

```
r2(config-if)#encapsulation frame-relay
```

```
r2(config-if)#no frame-relay inverse-arp
```

```
r2(config-if)#no arp frame-relay
```

```
r2(config-if)#ipv6 address 2023:1:1:11::3/64
```

```
r2(config-if)#frame-relay map ipv6 2023:1:1:11::3 203 broadcast
```

```
r2(config-if)#frame-relay map ipv6 FE80::213:1AFF:FE2F:380 203 broadcast
```

```
r2(config-if)#ipv6 ospf network point-to-point
```

```
r2(config-if)#ipv6 ospf 10 area 0
```

### (3) R3 初始配置:

```
r3(config)#ipv6 unicast-routing
```

```
r3(config)#ipv6 router ospf 10
```

```
r3(config-rtr)#router-id 3.3.3.3
```

```
r3(config)#interface loopback 0
```

```
r3(config-if)#ipv6 address 2033:3:3:33::3/64
```

```
r3(config-if)#ipv6 ospf network point-to-point
```

```
r3(config-if)#ipv6 ospf 10 area 0
```

```
r3(config)#int s0/0
```

```
r3(config-if)#encapsulation frame-relay
```

```
r3(config-if)#no frame-relay inverse-arp
```

```
r3(config-if)#no arp frame-relay
```

```
r3(config-if)#ipv6 address 2023:1:1:11::3/64
```

```
r3(config-if)#frame-relay map ipv6 2023:1:1:11::2 302 broadcast
```

```
r3(config-if)#frame-relay map ipv6 FE80::213:1AFF:FE2F:1200 302 broadcast
```

```
r3(config-if)#ipv6 ospf network point-to-point
```

```
r3(config-if)#ipv6 ospf 10 area 0
```

## 2. 开启 IPv6 组播

(1) 在 R1 上开启 IPv6 组播

```
r1(config)#ipv6 multicast-routing
```

(2) 在 R2 上开启 IPv6 组播

```
r2(config)#ipv6 multicast-routing
```

(3) 在 R3 上开启 IPv6 组播

```
r3(config)#ipv6 multicast-routing
```

## 3. 配置 IPv6 PIM

**说明：**在配置 IPv6 PIM 时，当开启 IPv6 组播功能后，所有正常启用 IPv6 功能的接口自动开启 IPv6 PIM，所以 IPv6 PIM 无须手工配置。

(1) 查看 R1 上的 PIM 状态

```
r1#show ipv6 pim neighbor
```

Neighbor Address	Interface	Uptime	Expires	DR	pri	Bidir
------------------	-----------	--------	---------	----	-----	-------

FE80::213:1AFF:FE2F:1200	FastEthernet0/0	00:00:43	00:01:31	1	(DR)	B
--------------------------	-----------------	----------	----------	---	------	---

```
r1#
```

**说明：**由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

## (2) 查看 R2 上的 PIM 状态

```
r2#show ipv6 pim neighbor
```

Neighbor Address	Interface	Uptime	Expires	DR	pri	Bidir
FE80::212:D9FF:FEF9:C8A0	FastEthernet0/0	00:00:56	00:01:18	1		B
FE80::213:1AFF:FE2F:380	Serial0/0	00:00:55	00:01:28	10 (DR)		B

```
r2#
```

**说明：**由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

## (3) 查看 R3 上的 PIM 状态

```
r3#show ipv6 pim neighbor
```

Neighbor Address	Interface	Uptime	Expires	DR	pri	Bidir
FE80::213:1AFF:FE2F:1200	Serial0/0	00:01:09	00:01:37	1		B

```
r3#
```

**说明：**由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

## 4.配置 MLD

**说明：**因为 MLD 的功能同 IGMP，所以配置的目的为加入某个组。

### (1) 在 R1 上配置加入组 ff04::1

**说明：** 建议使用未分配的组播地址范围。

```
r1(config)#interface loopback 0
```

```
r1(config-if)#ipv6 mld join-group ff04::1
```

## 5.配置静态 RP

**说明：** 静态配置所有设备的 RP 为 R1 的 loopback 0

### (1) 在 R1 上配置静态 RP

```
r1(config)#ipv6 pim rp-address 2011:1:1:11::1
```

### (2) 在 R2 上配置静态 RP

```
r2(config)#ipv6 pim rp-address 2011:1:1:11::1
```

### (3) 在 R3 上配置静态 RP

```
r3(config)#ipv6 pim rp-address 2011:1:1:11::1
```

## 6.查看结果

### (1) 查看 R1 上的 RP 情况

```
r1#show ipv6 pim group-map ff04::
```

```
FF00::/8*
```

```
SM, RP: 2011:1:1:11::1
```

```
RPF: Tu2,2011:1:1:11::1 (us)
```

```
Info source: Static
```

```
Uptime: 00:01:15, Groups: 1
```

r1#

**说明：**由于已手工配置 RP，所以 RP 正常。

### (2) 查看 R2 上的 RP 情况

r2#show ipv6 pim group-map ff04::

FF00::/8\*

SM, RP: 2011:1:1:11::1

RPF: Fa0/0,FE80::212:D9FF:FEF9:C8A0

Info source: Static

Uptime: 00:01:40, Groups: 0

r2#

**说明：**由于已手工配置 RP，所以 RP 正常。

### (3) 查看 R3 上的 RP 情况

r3#show ipv6 pim group-map ff04::

FF00::/8\*

SM, RP: 2011:1:1:11::1

RPF: Se0/0,FE80::213:1AFF:FE2F:1200

Info source: Static

Uptime: 00:01:41, Groups: 0

r3

**说明：**由于已手工配置 RP，所以 RP 正常。

## 7.测试组播通信情况

### （1）测试 R1 的组播通信情况

r1#ping ff04::1

Output Interface: Loopback0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2011:1:1:11::1

Reply to request 0 received from 2011:1:1:11::1, 16 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/3/16 ms

5 multicast replies and 0 errors.

r1#

**说明：**在测试组播时，出口必须全部写出。从结果中可以看出，由于 PIM 已成功建立，RP 已正确学到，所以组播通信正常。

### （2）测试 R2 的组播通信情况



r2#ping ff04::1

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2012:1:1:11::2

Reply to request 0 received from 2011:1:1:11::1, 4 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

5 multicast replies and 0 errors.

r2#

**说明：**从结果中可以看出，由于 PIM 已成功建立，RP 已正确学到，所以组播通信正常。

### （3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Request 0 timed out

Request 1 timed out

Request 2 timed out

Request 3 timed out

Request 4 timed out

Success rate is 0 percent (0/5)

0 multicast replies and 0 errors.

r3#

**说明：**R3 的组播无法 ping 通，由于 PIM 的 DR 选举问题。原因请参见 IPv4 组播部分。

## 8. 解决组播通信问题

**说明：**由于 R3 与 R2 之间为多路访问，DR 位置错误，所以组播无法通信，切换 DR 位置以解决组播通信问题。

### （1）改 R2 为网络中的 DR

```
r2(config)#interface s0/0
```

```
r2(config-if)#ipv6 pim dr-priority 100
```

### （2）查看当前网络中 DR 情况

```
r3#show ipv6 pim neighbor
```

Neighbor Address	Interface	Uptime	Expires	DR	pri	Bidir
FE80::213:1AFF:FE2F:1200	Serial0/0	00:10:02	00:01:24	100	(DR)	B

r3#

**说明：**DR 已成功变为 R2。

### （3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Reply to request 0 received from 2011:1:1:11::1, 68 ms

Reply to request 0 received from 2011:1:1:11::1, 80 ms

Reply to request 1 received from 2011:1:1:11::1, 64 ms

Reply to request 1 received from 2011:1:1:11::1, 76 ms

Reply to request 2 received from 2011:1:1:11::1, 65 ms

Reply to request 2 received from 2011:1:1:11::1, 77 ms

Reply to request 3 received from 2011:1:1:11::1, 68 ms

Reply to request 3 received from 2011:1:1:11::1, 80 ms

Reply to request 4 received from 2011:1:1:11::1, 124 ms

Reply to request 4 received from 2011:1:1:11::1, 168 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/87/168 ms

10 multicast replies and 0 errors.

r3#

**说明：**修改 DR 后，R3 的组播通信正常。

## 配置 BSR

**说明：**前面通过手工静态配置 RP 来使组播正常通信，下面通过配置 BSR 来自动选举 RP。

配置 R1 的 loopback0 为 C-BSR 和 C-RP。

### 1.配置 C-BSR

#### (1) 配置 R1 的 loopback0 为 C-BSR

```
r1(config)#ipv6 pim bsr candidate bsr 2011:1:1:11::1
```

### 2.配置 C-RP

#### (1) 配置 R1 的 loopback0 为 C-RP

```
r1(config)#ipv6 pim bsr candidate rp 2011:1:1:11::1
```

### 3.查看结果

#### (1) 查看 R1 上的 RP 情况

```
r1#show ipv6 pim group-map ff04::
```

```
FF00::/8*
```

```
SM, RP: 2011:1:1:11::1
```

```
RPF: Tu2,2011:1:1:11::1 (us)
```

```
Info source: BSR From: 2011:1:1:11::1(00:01:31), Priority: 192
```

```
Uptime: 00:00:58, Groups: 1
```

r1#

**说明：**由于 BSR 配置正确，所以 RP 正常。

### (2) 查看 R2 上的 RP 情况

r2#show ipv6 pim group-map ff04::

FF00::/8\*

SM, RP: 2011:1:1:11::1

RPF: Fa0/0,FE80::212:D9FF:FEF9:C8A0

Info source: BSR From: 2011:1:1:11::1(00:02:20), Priority: 192

Uptime: 00:01:09, Groups: 1

r2#

**说明：**由于 BSR 配置正确，所以 RP 正常。

### (3) 查看 R3 上的 RP 情况

r3#show ipv6 pim group-map ff04::

FF00::/8\*

SM, RP: 2011:1:1:11::1

RPF: Se0/0,FE80::213:1AFF:FE2F:1200

Info source: BSR From: 2011:1:1:11::1(00:02:09), Priority: 192

Uptime: 00:01:20, Groups: 0

r3#

**说明：** 由于 BSR 配置正确，所以 RP 正常。

#### 4.测试组播通信情况

##### （1）测试 R1 的组播通信情况

r1#ping ff04::1

Output Interface: Loopback0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2011:1:1:11::1

Reply to request 0 received from 2011:1:1:11::1, 8 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/8 ms

5 multicast replies and 0 errors.

r1#

**说明：** RP 已正确学到，所以组播通信正常。

##### （2）测试 R2 的组播通信情况

r2#ping ff04::1

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2012:1:1:11::2

Reply to request 0 received from 2011:1:1:11::1, 12 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 8 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/4/12 ms

5 multicast replies and 0 errors.

r2#

**说明：** RP 已正确学到，所以组播通信正常。

### （3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Reply to request 0 received from 2011:1:1:11::1, 120 ms

Reply to request 0 received from 2011:1:1:11::1, 132 ms

Reply to request 1 received from 2011:1:1:11::1, 100 ms

Reply to request 1 received from 2011:1:1:11::1, 116 ms

Reply to request 1 received from 2011:1:1:11::1, 128 ms

Reply to request 2 received from 2011:1:1:11::1, 100 ms

Reply to request 2 received from 2011:1:1:11::1, 116 ms

Reply to request 3 received from 2011:1:1:11::1, 100 ms

Reply to request 3 received from 2011:1:1:11::1, 116 ms

Reply to request 4 received from 2011:1:1:11::1, 101 ms

Reply to request 4 received from 2011:1:1:11::1, 117 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 100/113/132 ms

11 multicast replies and 0 errors.

r3#

**说明：** RP 已正确学到，所以组播通信正常。

### IPv6 邻居发现协议（ND 协议）

由于 IPv6 ND 协议中，几乎所有有用功能均为自动开启，无须手工干预，所以在此不再详细解释 ND 协议的运行过程，详细内容请自行参考 Cisco 文档中 IOS 12.4 T 部分。