

目 录

第 1 章 路由协议概述	1-1
1.1 路由表	1-1
1.2 IP路由策略	1-2
1.2.1 IP路由策略介绍	1-2
1.2.2 IP路由策略配置	1-4
1.2.3 配置案例	1-6
1.2.4 排错帮助	1-7
第 2 章 静态路由	2-1
2.1 静态路由介绍	2-1
2.2 缺省路由介绍	2-1
2.3 静态路由配置	2-1
2.4 配置案例	2-2
第 3 章 RIP	3-1
3.1 RIP介绍	3-1
3.2 RIP配置	3-2
3.3 RIP案例	3-9
3.3.1 RIP典型案例	3-9
3.3.2 RIP VPN典型案例	3-10
3.3.3 RIP路由聚合功能典型案例	3-12
3.4 RIP排错帮助	3-13
第 4 章 RIPng	4-1
4.1 RIPng介绍	4-1
4.2 RIPng配置	4-2
4.3 RIPng典型案例	4-6
4.3.1 RIPng典型案例	4-6
4.3.2 RIPng路由聚合功能典型案例	4-8
4.4 RIPng排错帮助	4-9
第 5 章 OSPF	5-1

5.1 OSPF介绍.....	5-1
5.2 OSPF配置.....	5-3
5.3 OSPF案例.....	5-8
5.3.1 OSPF典型案例.....	5-8
5.3.2 OSPF VPN典型案例.....	5-16
5.4 OSPF排错帮助.....	5-17
第 6 章 OSPFv3.....	6-1
6.1 OSPFv3 介绍.....	6-1
6.2 OSPFv3 配置.....	6-3
6.3 OSPFv3 案例.....	6-7
6.4 OSPFv3 排错帮助.....	6-10
第 7 章 BGP.....	7-1
7.1 BGP介绍.....	7-1
7.2 BGP配置.....	7-3
7.3 BGP典型案例.....	7-14
7.3.1 案例一：BGP邻居配置.....	7-14
7.3.2 案例二：BGP聚合配置.....	7-16
7.3.3 案例三：配置BGP团体属性.....	7-16
7.3.4 案例四：BGP联盟配置.....	7-17
7.3.5 案例五：BGP路由反射器配置.....	7-19
7.3.6 案例六：BGP的MED设置.....	7-20
7.3.7 案例七：BGP VPN典型案例.....	7-22
7.4 BGP排错帮助.....	7-27
第 8 章 MBGP4+.....	8-1
8.1 MBGP4+简介.....	8-1
8.2 MBGP4+配置.....	8-1
8.3 MBGP4+案例.....	8-2
8.4 MBGP4+排错帮助.....	8-4
第 9 章 黑洞路由操作手册.....	9-1
9.1 黑洞路由介绍.....	9-1
9.2 IPv4 黑洞路由配置任务.....	9-1

9.3 IPv6 黑洞路由配置任务.....	9-1
9.4 黑洞路由举例	9-2
9.5 黑洞路由排错帮助	9-3
第 10 章 GRE隧道配置	10-1
10.1 GRE隧道介绍	10-1
10.2 GRE隧道基本配置	10-1
10.3 GRE隧道举例	10-3
10.4 GRE隧道引用业务环回组举例	10-6
10.5 GRE隧道排错帮助	10-10
第 11 章 ECMP配置	11-1
11.1 ECMP简介	11-1
11.2 ECMP配置任务序列.....	11-1
11.3 ECMP典型案例.....	11-2
11.3.1 静态路由实现ECMP	11-2
11.3.2 OSPF实现ECMP	11-3
第 12 章 BFD.....	12-1
12.1 BFD介绍	12-1
12.2 BFD配置任务序列.....	12-1
12.3 BFD举例	12-3
12.3.1 BFD与静态路由联动举例	12-3
12.3.2 BFD与RIP路由联动举例	12-4
12.3.3 BFD与VRRP联动举例	12-5
12.4 BFD排错帮助.....	12-6
第 13 章 BGP GR配置	13-1
13.1 GR简介	13-1
13.2 GR配置任务序列.....	13-2
13.3 GR典型案例.....	13-4
第 14 章 OSPF GR配置	14-1
14.1 OSPF GR介绍	14-1

14.2 OSPF GR基本配置	14-2
14.3 OSPF GR举例	14-3
14.4 OSPF GR排错帮助	14-4

第1章 路由协议概述

在 Internet 中，一台主机为了访问远端的另一台主机，必须通过一系列路由器或三层交换机选择一条合适的路径。

路由器或三层交换机都是通过 CPU 来计算路径，不同的是三层交换机将计算出来的路径添加到交换芯片中，由芯片进行线速转发；而路由器是将计算出来的路径保存在路由表和路由缓存区中，由 CPU 负责数据转发。可见，路由器和三层交换机都可以进行路径的选择，而三层交换机在数据转发上有比较强大的优势。下面简单描述一下三层交换机的路径选取的基本原理和方法。

在路径选取过程中，每台三层交换机只负责根据收到数据包的目的地址来选择一条合适的中间路径，然后将数据包传送给下一个三层交换机，直至路径上的最后一台三层交换机将数据包传送给目的主机。每台三层交换机所完成的将数据包传送到下一个三层交换机而选择的路径，就被称作路由。路由可以分为直连路由、静态路由和动态路由等几种。

直连路由是指到与该三层交换机直接相连网络的路径，三层交换机不用计算就可以获得。

静态路由是指人为指定的到某个网络或特定主机的路径，静态即不能随意更改。静态路由的优点是简单易配、稳定、限制非法的路由改变，便于实现负载分担，便于实现路由备份。但是，因为是人为设置，对于大型网络需要设置的路由过于庞大和复杂，因此不适用于中大型网络。

动态路由是指三层交换机根据启动的路由协议动态计算到某个网络或特定主机的路径。如果路径中下一跳三层交换机不可达，三层交换机能够自动丢弃通过该三层交换机的路径，选择通过其它三层交换机的路径。

动态路由协议一般分为两类：内部网关路由协议（IGP）和外部网关路由协议（EGP）。内部网关路由协议（IGP）是用来计算到某个自治系统内目的路由的协议。三层交换机支持的内部网关动态路由协议有 RIP 和 OSPF 路由协议，可以根据需要配置 RIP 和 OSPF 路由协议。三层交换机支持同时运行多个内部网关动态路由协议，也可以在某个动态路由协议中重新引入其它动态路由协议和静态路由从而将多个路由协议联系起来。

外部网关路由协议是用来在不同自治系统之间交换路由信息，比如 BGP 协议。目前，本公司三层交换机支持的外部网关路由协议有 BGP-4、BGP4+等。

1.1 路由表

如前所述，三层交换机主要用于建立当前三层交换机到达某个网络或特定主机的路由，并根据路由转发数据包。每台三层交换机都建立有一张路由表，记录着该三层交换机使用的所有路由。路由表中每条路由项都指明了数据包到某子网或主机应该通过三层交换机的哪个物理端口发送，方可达到目的主机或到目的主机路径的下一台三层交换机。

路由表中包含下列一些主要的内容：

- ☞ 目的地址：用于标识 IP 数据包的目的地址或目的网络
- ☞ 网络掩码：与目的地址一起标识目的主机或三层交换机所在网段的网段地址。网络掩码由若干个连续的“1”构成，通常以点分十进制标识（一般为 1 到 4 个 255 组成的地址）。将目的地址和网络掩码“与”后，可以得到目的主机或三层交换机所在网段的网络地址。例如，目的地址为 200.1.1.1，掩码为 255.255.255.0 的主机或三层交换机所在网段的网络地址为 200.1.1.0。
- ☞ 输出接口：指明 IP 数据包将从该三层交换机的哪个接口转发。
- ☞ 下一台三层交换机（下一跳）IP 地址：指明 IP 数据包将经由的下一台三层交换机。
- ☞ 路由项优先级：针对同一目的地，可能存在不同下一跳的若干条路由。这些路由可能是不同的动态路由协议发现的，也可能是人为配置的静态路由。优先级高的（数值小）将成为当前的最优路由。用户可以配置到同一目的地优先级不同的多条路由，三层交换机将按优先级顺序选取唯一的一条路由用于 IP 数据包转发。

为了不使路由表过于庞大，可以设置一条缺省路由。一旦查找路由表失败后，就选择缺省路由转发数据包。

三层交换机支持的各种路由协议及其发现路由的缺省优先级如下表表示：

路由协议或路由种类	优先级缺省值
直连路由	0
OSPF	110
静态路由（static）	1
RIP	120
OSPF ASE	150
IBGP	200
EBGP	20
未知路由	255

1.2 IP路由策略

1.2.1 IP路由策略介绍

路由器在发布与接收路由信息时，可能需要实施一些策略，以便对路由信息进行过滤，比如只接收或发布一部分满足给定条件的路由信息；一种路由协议（如 RIP）可能需要引入（redistribute）其它的路由协议如（OSPF）发现的路由信息,从而丰富自己的路由知识；路由器在引入其它路由协议的路由信息时，可能需要只引入一部分满足条件的路由信息，并对所引入的路由信息的某些属性进行设置，以使其满足本协议的要求。

为实现路由策略，首先要定义将要实施路由策略的路由信息的特征，即定义一组匹配规则，可以以路由信息中的不同属性作为匹配依据进行设置，如目的地址、发布路由信息的路由器地址等。匹配规则可以预先设置好，然后再将它们应用于路由的发布接收和引入等过程的路由策略中。

在系统中提供了 route-map、acl、as-path、community-list 和 ip-prefix 五种过滤器供

路由协议引用，下面对各种过滤器逐个进行介绍：

1. route-map

用于匹配给定路由信息的某些属性，并在条件满足后对该路由信息的某些属性进行设置。

route-map 用于控制和改变路由信息，也可以控制路由之间的重新分配。一个 **route-map** 包含一系列 **match** 和 **set** 命令，**match** 命令指定需要满足的条件，**set** 命令则指明了当 **match** 条件匹配时要采取的相应动作。**route-map** 也用于控制不同路由进程之间的路由发布。**route-map** 还可用于策略路由，使报文选择不同的路径而非最短路径。

一组 **match** 和 **set** 子句组成一个节点，一个 **route-map** 可以由多个节点构成，每个节点是进行匹配测试的一个单元。节点间依据序列号 **sequence-number** 进行匹配。**match** 子句定义匹配规则，匹配对象是路由信息的一些属性。同一节点中的不同 **match** 子句是“与”的关系，只有满足节点内所有 **match** 子句指定的匹配条件，才能通过该节点的匹配测试。**set** 子句指定动作，也就是在通过节点的匹配测试后所执行的动作—对路由信息的一些属性进行设置。

一个 **route-map** 的不同节点间是“或”的关系，系统依次检查 **route-map** 的各个节点，如果通过了 **route-map** 的某一节点，就意味着通过该 **route-map** 的匹配测试，不进入下一个节点的测试。

2. 访问控制列表acl

ACL (Access Control Lists)是交换机实现的一种数据包过滤机制，通过允许或拒绝特定的数据包进出网络，交换机可以对网络访问进行控制，有效保证网络的安全运行。用户可以基于报文中的特定信息制定一组规则（**rule**），每条规则都描述了对匹配一定信息的数据包所采取的动作：允许通过（**permit**）或拒绝通过（**deny**）。用户可以把这些规则应用到特定交换机端口的入口或出口方向，这样特定端口上特定方向的数据流就必须依照指定的 **ACL** 规则进出交换机。请参考《**ACL 配置**》一章。

3. 前缀列表ip-prefix

前缀列表 **ip-prefix** 的作用类似于 **acl**，但比它更为灵活，且更易于为用户理解。**ip-prefix** 在应用于路由信息的过滤时，其匹配对象为路由信息的目的地址信息域。

一个 **ip-prefix** 由前缀列表名标识。每个前缀列表可以包含多个表项，每个表项可以独立指定一个网络前缀形式的匹配范围，并用一个 **sequence-number** 来标识，**sequence-number** 指明了在 **ip-prefix** 中进行匹配检查的顺序。

在匹配的过程中，交换机按升序依次检查由 **sequence-number** 标识的各个表项，只要有某一表项满足条件，就意味着通过该 **ip-prefix** 的过滤（不会进入下一个表项的测试）。

4. 自治系统路径信息访问列表as-path

自治系统路径信息访问列表 **as-path** 仅用于 **BGP**。**BGP** 的路由信息包中，包含有一自治系统路径域（在 **BGP** 交换路由信息的过程中，路由信息经过的自治系统路径会记录在这个域中）。**as-path** 就是针对自治系统路径域指定匹配条件。

as-path 的有关配置请用户参考 **BGP** 配置中的 **ip as-path** 命令。

5. 团体属性列表community-list

团体属性列表 community-list, 仅用于 BGP。BGP 的路由信息包中包含一个 community 属性域, 用来标识一个团体。community-list 就是针对团体属性域指定匹配条件。

community-list 有关的配置请用户参考 BGP 中的 ip community-list 命令。

1.2.2 IP路由策略配置

IP 路由策略配置任务序列:

- 1、定义 route-map
- 2、定义 route-map 的 match 语句
- 3、定义 route-map 的 set 语句
- 4、定义地址前缀列表

1.定义 route-map

命令	解释
全局配置模式	
route-map <map_name> {deny permit} <sequence_num> no route-map <map_name> [{deny permit} <sequence_num>]	配置 route-map; 本命令的 no 操作为删除 route-map。

2.定义 route-map 的 match 语句

命令	解释
route-map 配置模式	
match as-path <list-name> no match as-path [<list-name>]	匹配 BGP 路由经过的自制系统 AS 路径访问列表; 本命令的 no 操作为删除匹配。
match community <community-list-name community-list-num > [exact-match] no match community [<community-list-name community-list-num > [exact-match]]	匹配一个团体属性访问列表; 本命令的 no 操作为删除匹配。
match interface <interface-name > no match interface [<interface-name >]	按接口进行匹配; 本命令的 no 操作为删除匹配。
match ip <address next-hop> <ip-acl-name ip-acl-num prefix-list list-name> no match ip <address next-hop> [<ip-acl-name ip-acl-num prefix-list [list-name]>]	对地址或下一跳进行匹配; 本命令的 no 操作为删除匹配。

match metric <metric-val > no match metric [<metric-val >]	对路由度量值进行匹配；本命令的 no 操作为删除匹配。
match origin <egp igp incomplete > no match origin [<egp igp incomplete >]	对路由源进行匹配；本命令的 no 操作为删除匹配。
match route-type external <type-1 type-2 > no match route-type external [<type-1 type-2 >]	对路由类型进行匹配；本命令的 no 操作为删除匹配。
match tag <tag-val > no match tag [<tag-val >]	对路由 tag 进行匹配；本命令的 no 操作为删除匹配。

3. 定义 route-map 的 set 语句

命令	解释
route-map 配置模式	
set aggregator as <as-number> <ip_addr> no set aggregator as [<as-number> <ip_addr>]	为 BGP 聚合者分配一个 AS 号；本命令的 no 操作为删除配置。
set as-path prepend <as-num> no set as-path prepend [<as-num>]	在 BGP 路由信息的 as-path 系列前加入指定的 AS 号；本命令的 no 操作为删除配置。
set atomic-aggregate no set atomic-aggregate	设置 BGP 原子聚合属性；本命令的 no 操作为删除配置。
set comm-list <community-list-name community-list-num > delete no set comm-list <community-list-name community-list-num > delete	删除 BGP 团体属性值；本命令的 no 操作为删除配置。
set community [AA:NN] [internet] [local-AS] [no-advertise] [no-export] [none] [additive] no set community [AA:NN] [internet] [local-AS] [no-advertise] [no-export] [none] [additive]	设置 BGP 团体属性值；本命令的 no 操作为删除配置。
set extcommunity <rt soo> <AA:NN> no set extcommunity <rt soo> [<AA:NN>]	设置 BGP 扩展团体属性；本命令的 no 操作为删除配置。
set ip next-hop <ip_addr> no set ip next-hop [<ip_addr>]	设置下一跳 IP 地址；本命令的 no 操作为删除配置。
set local-preference <pre_val> no set local-preference [<pre_val>]	设置本地优先级；本命令的 no 操作为删除配置。

set metric < +/- metric_val metric_val > no set metric [< +/- metric_val metric_val >]	设置路由度量值；本命令的 no 操作为删除配置。
set metric-type <type-1 type-2> no set metric-type [<type-1 type-2>]	设置 OSPF 度量类型；本命令的 no 操作为删除配置。
set origin <egp igp incomplete > no set origin [<egp igp incomplete >]	设置 BGP 路由源；本命令的 no 操作为删除配置。
set originator-id <ip_addr> no set originator-id [<ip_addr>]	设置路由起源者 ID；本命令的 no 操作为删除配置。
set tag <tag_val> no set tag [<tag_val>]	设置 OSPF 路由 tag 值；本命令的 no 操作为删除配置。
set vpnv4 next-hop <ip_addr> no set vpnv4 next-hop [<ip_addr>]	设置 BGP VPNv4 下一跳地址；本命令的 no 操作为删除配置。
set weight < weight_val > no set weight [< weight_val >]	设置 BGP 路由权重；本命令的 no 操作为删除配置。

4. 定义地址前缀列表

命令	解释
全局配置模式	
ip prefix-list <list_name> description <description> no ip prefix-list <list_name> description	对前缀列表进行描述；本命令的 no 操作为删除配置。
ip prefix-list <list_name> [seq <sequence_number>] <deny permit> < any ip_addr/mask_length [ge min_prefix_len] [le max_prefix_len]> no ip prefix-list <list_name> [seq <sequence_number>] [<deny permit> < any ip_addr/mask_length [ge min_prefix_len] [le max_prefix_len]>]	配置前缀列表；本命令的 no 操作为删除配置。

1.2.3 配置案例

下图是由四台三层交换机组成的一个网络，本例子示范如何通过 route-map 进行 BGP 的 as_path 属性的设置。各三层交换机之间运行 BGP 协议。对 Switch3 而言，网络 192.68.11.0/24 可以通过两条路径得知，一是以 AS_PATH 1 经 IBGP 得知（经过 Switch4），二是以 AS_PATH 2 经 EBGP 得知（经过 Switch2）。BGP 优选最短路径，因此 AS_PATH 1 经 IBGP 的路径被优选。如果希望 BGP 优选路径 2，走 EBGP 路径，则可以将两个额外的 AS 路径号添加到从 Switch1 送到 Switch4 的 AS_PATH 信息中，以改变 Switch3 到达

192.68.11.0/24 的决策。

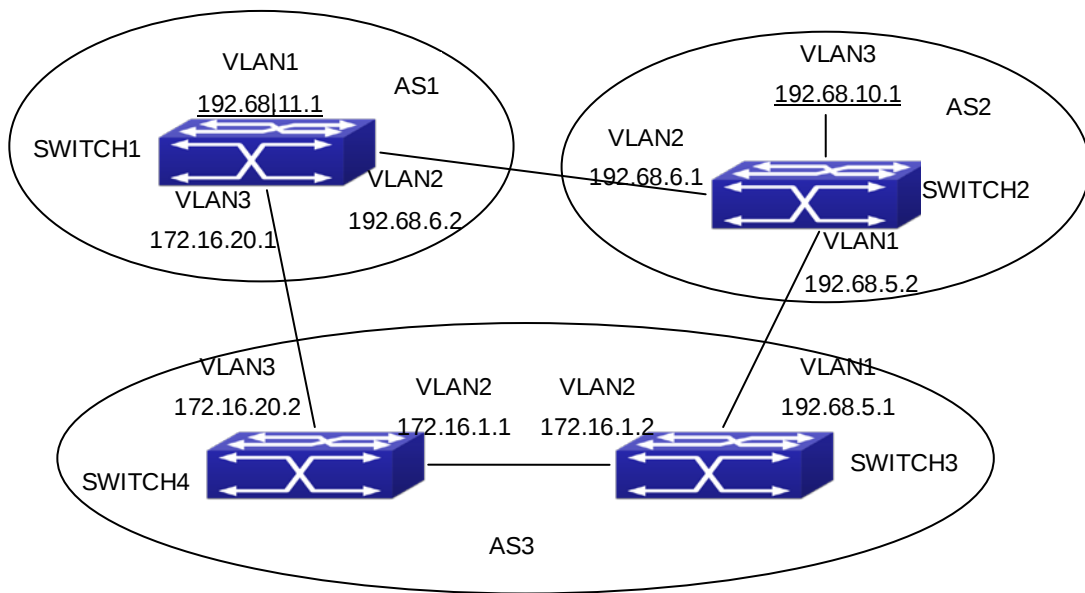


图 1-1 策略路由配置示意图

配置步骤：（此处仅列出 Switch1 的配置，其它交换机的配置省略）

三层交换机 Switch1 的配置

Switch1#config

Switch1(config)#router bgp 1

Switch1(config-router)#network 192.68.11.0 mask 255.255.255.0

Switch1(config-router)#neighbor 172.16.20.2 remote-as 3

Switch1(config-router)#neighbor 172.16.20.2 route-map AddAsNumbers out

Switch1(config-router)#neighbor 192.68.6.1 remote-as 2

Switch1(config-router)#exit

Switch1(config)#route-map AddAsNumbers permit 10

Switch1(config-route-map)#set as-path prepend 1 1

1.2.4 排错帮助

故障：路由协议运行正常的情况下无法实现路由信息学习

故障排除：检查如下几种错误

route-map 的各个节点中至少应该有一个节点的匹配模式是 permit 模式。当一个 route-map 用于路由信息过滤时，如果某路由信息没有通过任一节点的过滤，则认为该路由信息没有通过该 route-map 的过滤。当 route-map 的所有节点都是 deny 模式时，所有路由信息都不会通过该 route-map 的过滤。

地址前缀列表的各个表项中至少应该有一个表项的匹配模式是 permit 模式。deny 模式的表项可以先被定义，以快速的过滤掉不符合条件的路由信息。但如果所有表项都是 deny

模式，则任何路由都不会通过该地址前缀列表的过滤。可以在定义多条 **deny** 模式的表项后定义一条 **permit 0.0.0.0/0 le 32** 的表项，以允许其它所有路由信息通过。如果不指定 **less-equal 32** 将只匹配缺省路由。

第2章 静态路由

2.1 静态路由介绍

如前所述，静态路由是指人为指定的到某个网络或特定主机的路径。静态路由的优点是简单易配、稳定、能够禁止非法的路由改变，同时便于实现负载分担和路由备份。但是，它也存在许多缺点。静态路由是静态的，一旦网络发生故障不能自动修改路由，必须人为参与配置，不适用于中大型网络。

静态路由主要应用于两种情况：1）对于稳固的网络，为减少路由选择和路由数据流的负载，可使用静态路由。例如，到 STUB 网络的路由可采用静态路由。2）为实现路由备份，可以使用静态路由（在备份线路配置静态路由，路由优先级低于主线路）。

静态路由与动态路由可以同时存在，三层交换机根据路由协议优先级的不同，选用优先级最高的路由。同时，在动态路由中也可以通过重新引入静态路由的方式（redistribute），将静态路由加入到动态路由中，并根据需要改变引入静态路由的优先级。

2.2 缺省路由介绍

缺省路由也是一种静态路由，它是在没有找到任何匹配路由时才使用的路由。在路由表中，缺省路由的表示形式为目的地址为 0.0.0.0，网络掩码也为 0.0.0.0 的路由。如果路由表中没有数据包所要到达的目的地，也没有缺省路由，则丢弃该数据包，并向源地址返回一个 ICMP 数据报指出此目的地址或网络的不可达。

2.3 静态路由配置

静态路由配置任务序列：

- 1. 静态路由配置
- 2. VRF 配置

1.静态路由配置

命令	解释
全局配置模式	

<pre>ip route {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} {<gateway-address> <gateway-interface>} [<distance>] no ip route {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} [<gateway-address> <gateway-interface>] [<distance>]</pre>	配置静态路由；本命令的 no 操作作为删除静态路由。
---	----------------------------

2. VRF 配置

命令	解释
全局配置模式	
<pre>ip route vrf <name> {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} {<gateway-address> <gateway-interface>} [<distance>] no ip route vrf <name> {<ip-prefix> <mask> <ip-prefix>/<prefix-length>} [<gateway-address> <gateway-interface>] [<distance>]</pre>	配置静态路由；本命令的 no 操作作为删除静态路由。

2.4 配置案例

下图是由三台三层交换机组成的一个简单的网络，各三层交换机和 PC 机 IP 地址的网络掩码均为 255.255.255.0， Switch-1 与 Switch-3 之间都通过配置静态路由以使 PC1 和 PC3 之间进行通信，PC3 到 PC2 之间的通信通过在 Switch-3 上配置到 Switch-2 的静态路由来实现，PC2 到 PC3 之间的通信通过在 Switch-2 上配置缺省路由来实现。

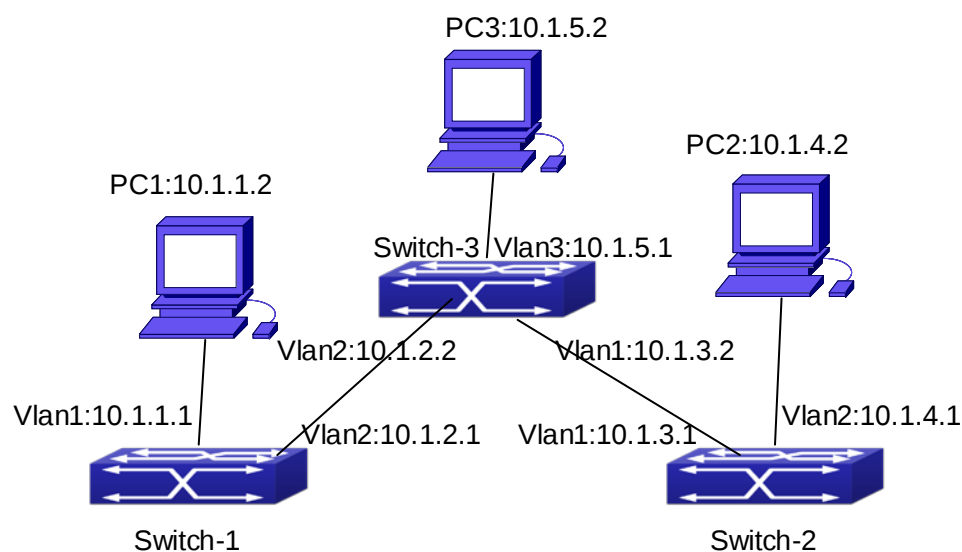


图 2-1 静态路由示意图

配置步骤:

三层交换机 Switch-1 的配置

```
Switch#config
```

```
Switch(config)#ip route 10.1.5.0 255.255.255.0 10.1.2.2
```

三层交换机 Switch-3 的配置

```
Switch#config
```

下一跳采用对端 IP 地址

```
Switch(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1
```

下一跳采用对端 IP 地址

```
Switch(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1
```

三层交换机 Switch-2 的配置

```
Switch#config
```

```
Switch(config)#ip route 0.0.0.0 0.0.0.0 10.1.3.2
```

这样，PC1 与 PC3、PC2 与 PC3 之间都可以 PING 通。

第3章 RIP

3.1 RIP介绍

RIP协议最早在ARPANET网络中使用，专门用于小型简单网络中。RIP协议是基于Bellman-Ford算法的距离向量路由协议。运行距离向量协议的网络设备定期向相邻设备发送两种信息：

- 到达目的网络所经过的跳数，即使用的度（metric），或者通过网络的数量。
- 下一跳是什么，或者达到目的网络要使用的方向（向量）。

距离向量三层交换机定期向相邻的三层交换机发送它们的整个路由选择表。三层交换机在从相邻三层交换机接收到的信息的基础之上建立自己的路由选择信息表。然后，将信息传递到它的相邻三层交换机。结果是路由选择表是在第二手信息的基础上建立的，距离代价超过15跳的路由将被视为不可达。

RIP协议是可选路由协议，它是基于UDP的协议，使用RIP的每个主机在UDP的520端口上发送和接收数据报。所有运行RIP协议的三层交换机，每隔30秒向所有邻居三层交换机发送路由表更新信息。如果180秒内没有收到来自对端的信息，那么认为该设备崩溃或相连的网络不可达。但到该三层交换机的路由还将在路由表中保持120秒，然后才被删除。

由于运行RIP协议的三层交换机使用第二手信息建立路由表，因此至少会遇到一个问题——无穷记数问题。对于一个运行RIP路由协议的网络，当某条RIP路由变为不可达，RIP三层交换机通常不会立刻发送路由更新报文，而是等待到达周期更新时间间隔（每30秒）时才发送有该路由信息的更新数据报。如果在没有收到更新报文之前，邻居向该三层交换机发送了一个包含邻居三层交换机自身路由表信息的数据报，那么会引起“无穷记数”现象，即出现到不可达三层交换机的路由选择度量固定递增的现象。这大大影响了路由选择、路由汇聚时间。

为了避免“无穷记数”现象，RIP协议提供了“水平分割”和“触发更新”等机制来解决路由循环问题。“水平分割”的原理是避免向网关发送从该网关学习到的路由，它包括“简单水平分割”——删除从邻居网关学习到的、将发送给该邻居网关的路由，和“逆向毒性水平分割”——不但在更新包中删除上述路由，而且将这些路由的代价设置为无穷。“触发更新”机制定义无论网关何时改变路由度量，都立即更新数据报并以广播形式发送出去，而不考虑30秒更新定时器的状态。

RIP协议包含版本1和版本2两个版本：RFC1058中介绍了RIP-I协议；RFC2453中介绍了RIP-II协议，同时兼容RFC1723和RFC1388。RIP-I采用发送广播数据报的方式发送路由更新数据报，它不支持子网掩码和认证。RIP-I的数据报中有一些域是不使用的，要求保证是全“0”，因此如果使用RIP-I应该进行全“0”域检查，如果这些域非“0”则丢弃此RIP-I数据报。RIP-II版本比RIP-I版本完善，它采用发送组播数据报的方式发送路由更新数据报（组播地址为224.0.0.9），它增加了子网掩码域和RIP验证域（支持简单明文密码和MD5密码验证），支持可变长子网掩码。RIP-II使用了RIP-I中的部分全“0”域，因

此无需进行全“0”域检查。DCRS 系列三层交换机缺省情况下是组播发送 RIP-II 数据报，接收 RIP-I 和 RIP-II 数据报。

每个运行 RIP 协议的三层交换机都有一个路由数据库，数据库中包含了该三层交换机所有可达目的地的路由项，并以此建立路由表。当 RIP 三层交换机向其相邻设备发送路由更新数据报时，路由更新数据报中包含了该三层交换机依据路由数据库建立的整个路由表。因此，对于较大的网络系统，每台三层交换机需要传输和处理的路由数据量很大、负担重，从而大大影响网络性能。

同时，RIP 协议支持将其它路由协议发现的路由信息引入到路由表中。也可以在 PE 路由器上作为与 CE 交换路由信息的协议，支持 VPN 路由/转发实例。

RIP 协议运行过程描述如下：

1. 启动 RIP，以广播形式向其相邻三层交换机发送请求数据报，相邻设备收到请求数据报后，响应该请求，并回送包含本地路由信息的响应数据报。
2. 三层交换机接收到响应数据报后，修改本地路由表，同时向相邻设备发送触发更新数据报，广播路由更新信息。相邻三层交换机接收到触发更新数据报之后，向其相邻的三层交换机发送触发更新数据报。经过一连串触发更新报的广播之后，各三层交换机都得到并保持最新的路由信息。

同时，RIP 三层交换机每隔 30 秒向其相邻设备广播本地路由表。相邻设备在接收到数据报之后，对本地路由进行维护，选择出最佳路由并向其各自的设备广播更新信息，使更新的路由最终达到全局有效。另外，RIP 采用超时机制对过时的路由进行超时处理，即三层交换机在一定时间间隔内（invalid 定时器间隔）没有收到来自某邻居的周期更新数据报，则认为来自该三层交换机的路由为无效路由，然后该路由再在路由表中保留一定时间间隔（holddown 定时器间隔），最后删除该路由。

3.2 RIP配置

RIP 配置任务序列：

1. 启动 RIP 协议（必须）
 - （1）启动 RIP 模块/关闭 RIP 模块
 - （2）配置运行 RIP 协议的网段
2. 配置 RIP 协议参数（可选）
 - （1）配置 RIP 发包机制
 - 1）配置 RIP 数据报的定点发送
 - 2）配置 RIP 接口广播
 - （2）配置 RIP 路由参数
 - 1）配置引入路由（缺省路由权值、配置 RIP 中引入其它协议的路由）
 - 2）配置接口的验证模式及密码
 - 3）配置路由偏移
 - 4）配置应用路由过滤

- 5) 配置水平分割
- (3) 配置 RIP 协议其它参数
 - 1) 配置 RIP 路由管理距离
 - 2) 配置路由表中 RIP 路由的数目限制
 - 3) 配置 RIP 更新、超时、抑制等计时器时间
 - 4) 配置 RIP UDP 接收缓冲区大小
3. 配置 RIP-I/RIP-II 模式切换
 - (1) 配置所有接口使用的 RIP 版本
 - (2) 配置接口发送/接收的 RIP 版本
 - (3) 配置接口是否发送/接收 RIP 数据报
4. 删除 RIP 路由表中的特定路由
5. 配置 RIP VPN 命令
6. 配置 RIP 路由聚合
 - (1) 配置 IPv4 路由器模式的聚合路由
 - (2) 配置 IPv4 接口配置模式的聚合路由
 - (3) 显示 IPv4 聚合路由信息
7. 配置 RIP 引入不同进程 OSPF 路由
 - (1) 启动 RIP 引入不同进程 OSPF 路由功能
 - (2) 显示和调试 RIP 引入不同进程 OSPF 路由功能相关信息
8. 配置 RIP 的 VRF 地址族模式
 - (1) 启动 RIP 模块/关闭 RIP 模块
 - (2) 引用 VRF 地址族

1. 启动 RIP 协议

在系统中运行 RIP 路由协议的基本配置很简单，通常只需打开 RIP 开关、并且配置运行 RIP 的网段，即按 RIP 缺省配置发送和接收 RIP 数据报。如果需要可以切换发送、接收 RIP 数据报的版本，允许/禁止发送、接收 RIP 数据报，参考 3。

命令	解释
全局配置模式	
router rip no router rip	打开 RIP 协议；本命令的 no 操作关闭 RIP 协议。
路由器与地址族配置模式	
network <A.B.C.D/M ifname vlan> no network <A.B.C.D/M ifname vlan>	设定运行 RIP 协议的网段；本命令的 no 操作为删除运行 RIP 协议的网段。

2. 配置 RIP 协议参数

- (1) 配置 RIP 发包机制

1) 配置 RIP 数据报的定点发送

2) 配置 RIP 接口广播

命令	解释
路由器配置模式	
neighbor <A.B.C.D> no neighbor <A.B.C.D>	指定需要定点发送的邻居路由器的 IP 地址；本命令的 no 操作取消指定的路由器。
passive-interface<ifname vlan> no passive-interface<ifname vlan>	指示 RIP 三层交换机在指定的接口上阻塞 RIP 广播，只能向配置了 neighbor 的三层交换机之间发送 RIP 数据包。本命令的 no 操作取消该功能。

(2) 配置 RIP 路由参数

1) 配置引入路由（缺省路由权值、配置 RIP 中引入其它协议的路由）

命令	解释
路由器配置模式	
default-metric <value> no default-metric	设定引入路由的缺省路由权值；本命令的 no 操作恢复缺省设置 1。
redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>] no redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>]	在 RIP 数据报中引入从其它路由协议引入的路由；本命令的 no 操作取消引入的相应协议的路由。
default-information originate no default-information originate	产生一条默认路由到 RIP 协议中；no 操作取消该特性。

2) 配置接口的认证模式及密码

命令	解释
接口配置模式	
ip rip authentication mode { text md5} no ip rip authentication mode [text md5]	设置认证使用的类型；本命令的 no 操作设置为取消该认证操作。
ip rip authentication string <text> no ip rip authentication string	设置认证使用的密钥；本命令的 no 操作不使用密钥。

ip rip authentication key-chain <name-of-chain> no ip rip authentication key-chain [<name-of-chain>]	设置认证使用的密钥链，本命令的 no 操作不使用密钥链。
全局模式	
key chain <name-of-chain> no key chain <name-of-chain>	进入 keychain 模式，并且配置一个 key chain; 本命令的 no 操作删除该 key chain。
Keychain模式	
key <keyid> no key <keyid>	进入 keychain-key 模式，并且配置 key chain 中的一个 key; 本命令的 no 操作删除一个 key。
Keychain-key模式	
key-string <text> no key-string <text>	配置被 key 用的密码; 本命令的 no 操作删除该密码。
accept-lifetime <start-time> {<end-time> duration<seconds> infinite} no accept-lifetime	配置 key chain 上的一个 key 接受为合法的时间; no 操作删除它。
send-lifetime <start-time> {<end-time> duration<seconds> infinite} no send-lifetime	配置 key chain 上的一个 key 可以发送的时间; no 操作删除 send-lifetime。

3) 配置路由偏移

命令	解释
路由器配置模式	
offset-list <access-list-number access-list-name> {in out }<number>[<ifname>] no offset-list <access-list-number access-list-name> {in out }<number>[<ifname>]	设定接口发送和接收 RIP 数据报时给路由的度量一个偏移量; 本命令的 no 操作移去偏移列表。

4) 配置应用路由过滤

命令	解释
路由器配置模式	

distribute-list {< access-list-number access-list-name > prefix<prefix-list- name>}{in out} [<ifname>] no distribute-list {< access-list-number access-list-name > prefix<prefix-list- name>}{in out} [<ifname>]	配置应用访问列表和前缀列表来过滤路由。本命令的 no 操作配置不使用访问列表和前缀列表。
---	--

5) 配置水平分割

命令	解释
接口配置模式	
ip rip split-horizon [poisoned] no ip rip split-horizon	配置接口发送数据报时进行水平分割操作，poisoned 表示带逆向毒化。No 操作取消水平分割操作

(3) 配置 RIP 协议其它参数

- 1) 配置 RIP 路由优先级
- 2) 配置路由表中 RIP 路由的数目限制
- 3) 配置 RIP 更新、超时、抑制等计时器时间
- 4) 配置 RIP UDP 接收缓冲区大小

命令	解释
路由器配置模式	
distance <number> [<A.B.C.D/M>] [<access-list-name access-list-number >] no distance [<A.B.C.D/M>]	指定 RIP 协议的路由管理距离；本命令的 no 操作恢复缺省值 120。
maximum-prefix <maximum-prefix>[<threshold>] no maximum-prefix <maximum-prefix > no maximum-prefix	配置路由表中 RIP 路由的最大条数；本命令的 no 操作取消对路由条数的限制。
timers basic <update> <invalid> <garbage> no timers basic	调整 RIP 计时器更新、期满、垃圾收集的时间；本命令的 no 操作回复缺省设置。
recv-buffer-size <size> no recv-buffer-size	本命令配置 RIP 的 UDP 接收缓冲区大小；no 操作恢复系统缺省值。

3. 配置 RIP-I/RIP-II 模式切换

(1) 配置所有接口使用的 RIP 版本

命令	解释
RIP 协议配置模式	
version { 1 2 } no version	设置所有三层交换机接口发送/接收 RIP 数据报的版本；no 操作恢复缺省设置，即版本 2。

(2) 配置接口发送/接收的 RIP 版本

(3) 配置接口是否发送/接收 RIP 数据报

命令	解释
接口配置模式	
ip rip send version { 1 1-compatible 2 } no ip rip send version	设置接口发送 RIP 数据报版本；本命令的 no 操作设置接口使用 version 命令设置的版本。
ip rip receive version { 1 2 } no ip rip receive version	设置接口接收 RIP 数据报版本；本命令的 no 操作设置接口使用 version 命令设置的版本。
ip rip receive-packet no ip rip receive-packet	设定在接口上接收 RIP 数据报；本命令的 no 操作关闭本接口的 RIP 数据报接收。
ip rip send-packet no ip rip send-packet	设定在接口上发送 RIP 数据报；本命令的 no 操作关闭本接口的 RIP 数据报发送。

4. 删除 RIP 路由表中的特定路由

命令	解释
特权配置模式	
clear ip rip route {<A.B.C.D/M> kernel static connected rip ospf isis bgp all}	本命令删除 RIP 路由表中的特定路由。

5. 配置 RIP VPN 命令

命令	解释
路由器配置模式	
address-family ipv4 vrf <vrf-name> no address-family ipv4 vrf <vrf-name>	本命令为配置在 PE 路由器上的 VRF 配置 RIP 地址族。No 操作删除配置的地址族。
地址族配置模式	
exit-address-family	本命令退出地址族模式。

6. 配置 RIP 路由聚合

(1) 配置 IPv4 全局聚合路由

命令	解释
路由器配置模式	
ip rip aggregate-address A.B.C.D/M no ip rip aggregate-address A.B.C.D/M	配置或取消 IPv4 全局聚合路由。

(2) 配置 IPv4 接口聚合路由

命令	解释
接口配置模式	
ip rip aggregate-address A.B.C.D/M no ip rip aggregate-address A.B.C.D/M	配置或取消 IPv4 接口聚合路由。

(3) 显示 IPv4 聚合路由信息

命令	解释
特权模式和配置模式	
show ip rip aggregate	显示聚合路由信息。

7. 配置 RIP 引入不同进程 OSPF 路由

(1) 启动 RIP 引入不同进程 OSPF 路由功能

命令	解释
Router rip 配置模式	
redistribute ospf [<process-id>] [metric<value>] [route-map<word>] no redistribute ospf [<process-id>]	启动或者关闭 RIP 引入其他 OSPF 进程路由功能。

(2) 显示和调试 RIP 引入不同进程 OSPF 路由功能相关信息

命令	解释
特权和配置模式	
show ip rip redistribute	显示 RIP 进程引入其它外部路由的配置信息。
特权模式	
debug rip redistribute message send no debug rip redistribute message send debug rip redistribute route receive no debug rip redistribute route receive	打开或关闭 RIP 引入其他 OSPF 进程路由的命令下发调试开关。 打开或关闭 RIP 进程收到 nsm 发来的路由信息的调试开关。

8. 配置 RIP 的 VRF 地址族模式

命令	解释
----	----

Router RIP 配置模式	
address-family ipv4 vrf <vrf-name> no address-family ipv4 vrf <vrf-name>	本命令为配置在 PE 路由器上的 VRF 配置 RIP 地址族。No 操作删除配置的地址族。
地址族配置模式	
exit-address-family	本命令退出地址族模式。

3.3 RIP案例

3.3.1 RIP典型案例

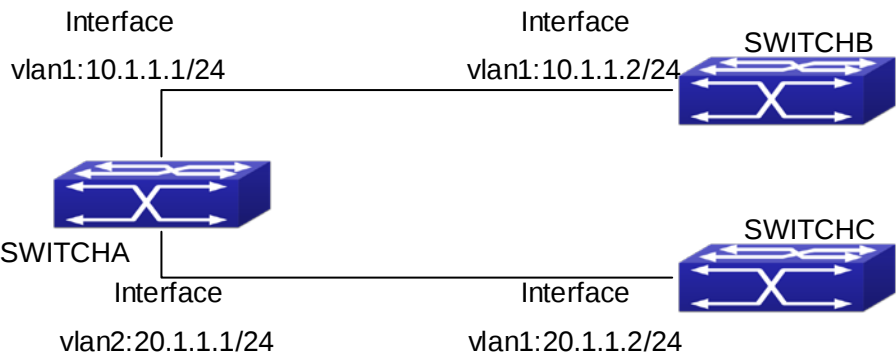


图 3-1 RIP 案例

图 3-1 为三台三层交换机组成的一个网络，三层交换机 SWITCHA 与三层交换机 SWITCHB 和三层交换机 SWITCHC 相连，三台三层交换机都运行 RIP 路由协议。设三层交换机 SWITCHA（interface vlan1: 10.1.1.1, interface vlan2: 20.1.1.1）只与三层交换机 SWITCHB（interface vlan1: 10.1.1.2）交流三层交换机更新信息，而不向三层交换机 SWITCHC（interface vlan1: 20.1.1.2）交流三层交换机更新信息。

三层交换机 SWITCHA、SWITCHB、SWITCHC 的配置分别如下：

a) 三层交换机 SwitchA:

配置 interface vlan1 的 IP 地址。

SwitchA#config

SwitchA(config)# interface vlan 1

SwitchA(Config-if-Vlan1)# ip address 10.1.1.1 255.255.255.0

SwitchA (config-if-Vlan1)#

配置 interface vlan2 的 IP 地址

```
SwitchA (config)# vlan 2
SwitchA (Config-Vlan2)# switchport interface ethernet 1/2
    Set the port Ethernet1/2 access vlan 2 successfully
SwitchA (Config-Vlan2)# exit
SwitchA (Config)# interface vlan 2
SwitchA (Config-if-Vlan2)# ip address 20.1.1.1 255.255.255.0
启动 RIP 协议，配置 RIP 网段
SwitchA(config)#router rip
SwitchA(config-router)#network vlan 1
SwitchA(config-router)#network vlan 2
SwitchA(config-router)#exit
配置接口 interface vlan 2 不向交换机 SwitchC 发送 RIP 报文
SwitchA(config)#router rip
SwitchA(config-router)#passive-interface vlan 2
SwitchA(config-router)#exit
SwitchA (config) #
b) 三层交换机 SwitchB:
配置 interface vlan1 的 IP 地址。
SwitchB#config
SwitchB(config)# interface vlan 1
SwitchB(Config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (Config-if-Vlan1)#exit
启动 RIP 协议，配置 RIP 网段
SwitchB(config)#router rip
SwitchB(config-router)#network vlan 1
SwitchB(config-router)#exit
c) 三层交换机 SwitchC:
配置 interface vlan1 的 IP 地址。
SwitchC#config
SwitchC(config)# interface vlan 1
SwitchC(Config-if-Vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC (Config-if-Vlan1)#exit
启动 RIP 协议，配置 RIP 网段
SwitchC(config)#router rip
SwitchC(config-router)#network vlan 1
SwitchC(config-router)#exit
```

3.3.2 RIP VPN典型案例

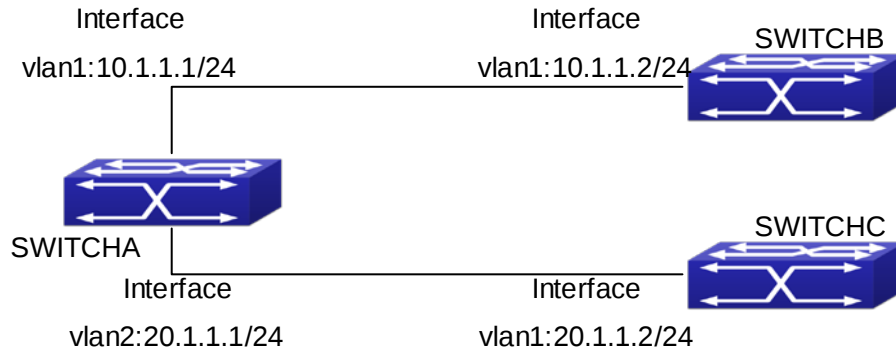


图 3-2 RIP VPN 案例

图 3-2 为三台三层交换机组成的网络。交换机 SwitchA 充当 PE，交换机 SWITCHB 和交换机 SwitchC 充当 CE1 与 CE2。PE 通过接口 vlan1 与 vlan2 分别与 CE1 和 CE2 相连。PE 与 CE 之间通过 RIP 协议交换路由信息

a) 作为 PE 的三层交换机 SwitchA:

配置 VPN 路由/转发实例 vpnb 和 vpnc

```
SwitchA#config
```

```
SwitchA(config)#ip vrf vpnb
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

```
SwitchA#(config)
```

```
SwitchA(config)#ip vrf vpnc
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

配置接口 vlan1 与 vlan2 分别与 vpnb 和 vpnc 相关联，并且配置 IP 地址

```
SwitchA(config)#in vlan1
```

```
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
```

```
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan1)#exit
```

```
SwitchA(config)#in vlan2
```

```
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc
```

```
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan2)#exit
```

分别配置与 vpnb 和 vpnc 相关联的 RIP 实例

```
SwitchA(config)#
```

```
SwitchA(config)#router rip
```

```
SwitchA(config-router)#address-family ipv4 vrf vpnb
```

```
SwitchA(config-router-af)#redistribute bgp
```

```
SwitchA(config-router-af)#network Vlan1
```

```
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router)#address-family ipv4 vrf vpnc
SwitchA(config-router-af)#redistribute bgp
SwitchA(config-router-af)#network Vlan2
SwitchA(config-router-af)#exit-address-family
```

```
SwitchA(config-router)#
```

b) CE1 三层交换机 SwitchB:

配置以太网口 E1/2 的 IP 地址。

```
SwitchB#config
```

```
SwitchB(config)# interface Vlan1
```

```
SwitchB(config-if-Vlan1)# ip address 10.1.1.2 255.255.255.0
```

```
SwitchB (config-if-Vlan1)#exit
```

启动 RIP 协议，配置 RIP 网段

```
SwitchB(config)#router rip
```

```
SwitchB(config-router-rip)#network Vlan1
```

```
SwitchB(config-router-rip)#exit
```

c) CE2 三层交换机 SwitchC:

配置以太网口 E1/2 的 IP 地址。

```
SwitchC#config
```

```
SwitchC(config)# interface Vlan1
```

```
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

```
SwitchC (config-if-vlan1)#exit
```

启动 RIP 协议，配置 RIP 网段

```
SwitchC(config)#router rip
```

```
SwitchC(config-router)#network Vlan1
```

```
SwitchC(config-router)#exit
```

3.3.3 RIP路由聚合功能典型案例

如下图的应用环境：

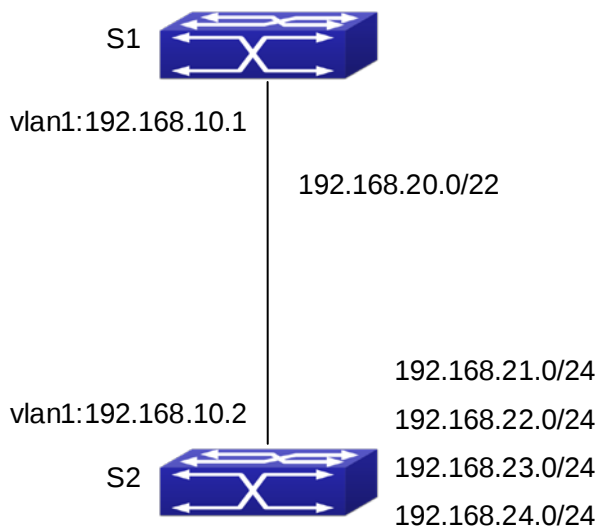


图 3-3 RIP 路由聚合典型应用环境

在上述网络拓扑图中，S2通过接口vlan1与S1相连，S2另外有四条子网路由，分别是：192.168.21.0/24，192.168.22.0/24，192.168.23.0/24，192.168.24.0/24。S2支持路由聚合，在S2的接口vlan1上配置聚合路由192.168.20.0/22后，通过vlan1给S1发送路由信息时，将四条子网路由聚合成一条聚合路由192.168.20.0/22，发送给S1，而不将子网发送给邻居。减小了S1的路由表规模，节省了内存。

S1配置序列：

```
S1(config)#router rip
S1(config-router) #network vlan 1
```

S2配置序列：

```
S2(config)#router rip
S2(config-router) #network vlan 1
S2(config-router) #exit
S2(config)#in vlan 1
S2(Config-if-Vlan1)# ip rip agg 192.168.20.0/22
```

3.4 RIP排错帮助

在配置、使用 RIP 协议时，可能会由于物理连接、配置错误等原因导致 RIP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后，先启动 RIP 协议（使用 router rip 命令）并且配置网段（使用 network 命令）再在相应接口配置 RIP 协议参数，如使用的是 RIP-I 还是 RIP-II 等；
- ☞ 接着，注意 RIP 协议的自身特点——RIP 三层交换机每隔 30 秒向它所有邻居三层交换机发送路由表更新信息，如果 180 秒内没有收到来自某三层交换机的信息，那么认为该三层交换机崩溃或相连的网络不可达，到该三层交换机的路由还将在路由表中保持

120 秒然后被删除。因此，如果删除某个 RIP 路由，需等待 300 秒后才能保证路由表中除去此路由项；

- ☞ 如果是在 PE 路由器上使用 RIP 协议与 CE 交换路由信息。首先应该创建相应的 VPN 路由/转发实例，将相关接口与之关联。然后进入 RIP 的地址族模式配置相应的参数如果仍无法解决 RIP 路由问题，那么请使用 debug rip 调试命令，然后将 3 分钟内的 debug 信息拷贝下来，发送给本公司技术服务中心。

第4章 RIPng

4.1 RIPng介绍

RIP协议最早在ARPANET网络中上使用，专门用于小型简单网络中。RIPng协议是基于Bellman-Ford算法的距离向量路由协议，是用于IPv6的RIP协议。运行距离向量协议的网络设备定期向相邻设备发送两种信息：

- 到达目的网络所经过的跳数，即使用的度（metric），或者通过网络的数量。
- 下一跳是什么，或者达到目的网络要使用的方向（向量）。

距离向量三层交换机定期向相邻的三层交换机发送它们的整个路由选择表。三层交换机在从相邻三层交换机接收到的信息的基础之上建立自己的路由选择信息表。然后，将信息传递到它的相邻三层交换机。结果是路由选择表是在第二手信息的基础上建立的，距离代价超过15跳的路由将被视为不可达。

RIPng协议是可选路由协议，它是基于UDP的协议，使用RIPng的每个主机在UDP的521端口上发送和接收数据报。所有运行RIPng协议的三层交换机，每隔30秒向所有邻居三层交换机发送路由表更新信息。如果180秒内没有收到来自对端的信息，那么认为该设备崩溃或相连的网络不可达。但到该三层交换机的路由还将在路由表中保持120秒，然后才被删除。

由于运行RIPng协议的三层交换机使用第二手信息建立路由表，因此至少会遇到一个问题——无穷记数问题。对于一个运行RIPng路由协议的网络，当某条RIPng路由变为不可达，RIPng三层交换机通常不会立刻发送路由更新报文，而是等待到达周期更新时间间隔（每30秒）时才发送有该路由信息的更新数据报。如果在没有收到更新报文之前，邻居向该三层交换机发送了一个包含邻居三层交换机自身路由表信息的数据报，那么会引起“无穷记数”现象，即出现到不可达三层交换机的路由选择度量固定递增的现象。这大大影响了路由选择、路由汇聚时间。

为了避免“无穷记数”现象，RIPng协议提供了“水平分割”和“触发更新”等机制来解决路由循环问题。“水平分割”的原理是避免向网关发送从该网关学习到的路由，它包括“简单水平分割”——删除从邻居网关学习到的、将发送给该邻居网关的路由，和“逆向毒性水平分割”——不但在更新包中删除上述路由，而且将这些路由的代价设置为无穷。“触发更新”机制定义无论网关何时改变路由度量，都立即将更新数据报并以广播形式发送出去，而不等待达到30秒更新时间间隔。

RIPng协议目前只有版本1一个版本：RFC2080中介绍了RIPng协议。RIPng采用发送组播数据报的方式发送路由更新数据报（组播地址为FF02::9）。

每个运行RIPng协议的三层交换机都有一个路由数据库，数据库中包含了该三层交换机所有可达目的地的路由项，并以此建立路由表。当RIPng三层交换机向其相邻设备发送路由更新数据报时，路由更新数据报中包含了该三层交换机依据路由数据库建立的整个路由

表。因此，对于较大的网络系统，每台三层交换机需要传输和处理的路由数据量很大、负担重，从而大大影响网络性能。

同时，RIPng 协议支持将其它 IPv6 路由协议发现的路由信息引入到路由表中。

RIPng 协议运行过程描述如下：

1. 启动 RIPng，以组播形式向其相邻三层交换机发送请求数据报，相邻设备收到请求数据报后，响应该请求，并回送包含本地路由信息的响应数据报。
2. 三层交换机接收到响应数据报后，修改本地路由表，同时向相邻设备发送触发更新数据报，广播路由更新信息。相邻三层交换机接收到触发更新数据报之后，向其相邻的三层交换机发送触发更新数据报。经过一连串触发更新报的广播之后，各三层交换机都得到并保持最新的路由信息。

同时，RIPng 三层交换机每隔 30 秒向其相邻设备广播本地路由表。相邻设备在接收到数据报之后，对本地路由进行维护，选择出最佳路由并向其各自的设备广播更新信息，使更新的路由最终达到全局有效。另外，RIPng 采用超时机制对过时的路由进行超时处理，即三层交换机在一定时间间隔内（invalid 定时器间隔）没有收到来自某邻居的周期更新数据报，则认为来自该三层交换机的路由为无效路由，然后该路由再在路由表中保留一定时间间隔（garbage collect 定时器间隔），最后删除该路由。

由于 IPv6 网络还在发展中，有时网络中存在不支持 IPv6 的网络环境，这就需要通过隧道进行 IPv6 的操作，所以我们的 RIPng 支持在配置隧道上的配置和运行，以 IPV4 单播的方式穿过不支持 IPv6 的网络。

4.2 RIPng配置

RIPng 配置任务序列：

1. 启动 RIPng 协议（必须）
 - （1） 启动/关闭 RIPng 协议
 - （2） 配置运行 RIPng 协议的网段
2. 配置 RIPng 协议参数（可选）
 - （1） 配置 RIPng 发包机制
 - 1) 配置 RIPng 数据报的定点发送
 - （2） 配置 RIPng 路由参数
 - 1) 配置引入路由（缺省路由权值、配置 RIPng 中引入其它协议的路由）
 - 2) 配置路由偏移
 - 3) 配置应用路由过滤
 - 4) 配置水平分割
3. 配置 RIPng 协议其它参数
 - （1） 配置 RIPng 更新、超时、抑制等计时器时间
4. 删除 RIPng 路由表中的特定路由
5. 配置 RIPng 路由聚合

- (1) 配置 IPv6 路由器模式的聚合路由
 - (2) 配置 IPv6 接口配置模式的聚合路由
 - (3) 显示 IPv6 聚合路由信息
6. 配置 RIPng 引入不同进程 OSPFv3 路由
- (1) 启动 RIPng 引入不同进程 OSPFv3 路由功能
 - (2) 显示和调试 RIPng 引入不同进程 OSPFv3 路由功能相关信息

1. 启动 RIPng 协议

在系统中运行 RIPng 路由协议的基本配置很简单，通常只需打开 RIPng 开关、并且配置运行 RIPng 的接口，即按 RIPng 缺省配置发送和接收 RIPng 数据报。

命令	解释
全局配置模式	
[no] router IPv6 rip	打开 RIPng 协议；本命令的 no 操作关闭 RIPng 协议。
接口配置模式	
[no] IPv6 router rip	设定运行 RIPng 协议的接口；本命令的 no 操作作为设定接口不运行 RIPng 协议。

2. 配置 RIPng 协议参数

(1) 配置 RIPng 发包机制

1) 配置 RIPng 数据报的定点发送

命令	解释
路由器配置模式	
[no] neighbor <IPv6-address> <ifname>	指定需要定点发送的邻居路由器的 IPv6 Link-local 地址和接口名；本命令的 no 操作取消指定的路由器。
[no] passive-interface <ifname>	指示 RIPng 三层交换机在指定的接口上阻塞 RIPng 组播，只能向配置了 neighbor 的三层交换机之间发送 RIPng 数据包；本命令的 no 操作取消该功能。

(2) 配置 RIPng 路由参数

1) 配置引入路由（缺省路由权值、配置 RIPng 中引入其它协议的路由）

命令	解释
路由器配置模式	
default-metric <value> no default-metric	设定引入路由的缺省路由权值；本命令的 no 操作恢复缺省设置 1。

[no] redistribute {kernel connected static ospf isis bgp} [metric<value>] [route-map<word>]	在 RIPng 数据报中引入从其它路由协议引入的路由；本命令的 no 操作取消引入的相应协议的路由。
[no] default-information originate	产生一条默认路由到 RIPng 协议中；no 操作取消该特性。

2) 配置路由偏移

命令	解释
路由器配置模式	
[no] offset-list <access-list-number access-list-name> {in out} <number> [<ifname>]	设定接口发送和接收 RIPng 数据报时给路由的度量一个偏移量；本命令的 no 操作移去偏移列表。

3) 配置应用路由过滤和路由聚合

命令	解释
路由器配置模式	
[no] distribute-list {< access-list-number access-list-name> prefix<prefix-list-name>} {in out} [<ifname>]	设定接口发送和接收 RIPng 数据报时过滤路由；本命令的 no 操作不配置路由过滤。
[no] aggregate-address <IPv6-address>	配置路由聚合，no 命令取消路由聚合。

4) 配置水平分割

命令	解释
接口配置模式	
IPv6 rip split-horizon [poisoned]	配置接口发送数据报时进行水平分割操作，POISONED 表示带逆向毒化。
no IPv6 rip split-horizon	取消水平分割操作。

3. 配置 RIPng 协议其它参数

(1) 配置 RIPng 更新、超时、抑制等计时器时间

命令	解释
路由器配置模式	

timers basic <update> <invalid> <garbage> no timers basic	调整 RIPng 计时器更新、期满、垃圾收集的时间；本命令的 no 操作回复缺省设置。
---	---

4. 删除 RIPng 路由表中的特定路由

命令	解释
特权配置模式	
clear IPv6 rip route {<IPv6-address> kernel static connected rip ospf isis bgp all}	本命令删除 RIPng 路由表中的特定路由。

5. 配置 RIPng 路由聚合

(1) 配置 IPv6 全局聚合路由

命令	解释
路由器配置模式	
ipv6 rip aggregate-address X:X::X:X/M no ipv6 rip aggregate-address X:X::X:X/M	配置或取消 IPv6 全局聚合路由。

(2) 配置 IPv6 接口聚合路由

命令	解释
接口配置模式	
ipv6 rip aggregate-address X:X::X:X/M no ipv6 rip aggregate-address X:X::X:X/M	配置或取消 IPv6 接口聚合路由。

(3) 显示 IPv6 聚合路由信息

命令	解释
特权模式和配置模式	
show ipv6 rip aggregate	显示 IPv6 聚合路由信息，如聚合的接口，度量，聚合的路由数，被聚合的次数。

6. 配置 RIPng 引入不同进程 OSPFv3 路由

(1) 启动 RIPng 引入不同进程 OSPFv3 路由功能

命令	解释
Router ipv6 rip 配置模式	

redistribute ospf [<process-tag>] [metric<value>] [route-map <word>] no redistribute ospf [<process-tag>]	启动或者关闭 RIPng 引入其他 OSPFv3 进程路由功能。
--	----------------------------------

(2) 显示和调试 RIPng 引入不同进程 OSPFv3 路由功能相关信息

命令	解释
特权配置模式	
show ipv6 rip redistribute	显示 RIPng 进程引入其它外部路由的配置信息。
特权模式	
debug ipv6 rip redistribute message send no debug ipv6 rip redistribute message send debug ipv6 rip redistribute route receive no debug ipv6 rip redistribute route receive	打开或关闭 RIPng 引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 RIPng 进程收到 nsm 发来的路由信息的调试开关。

4.3 RIPng典型案例

4.3.1 RIPng典型案例

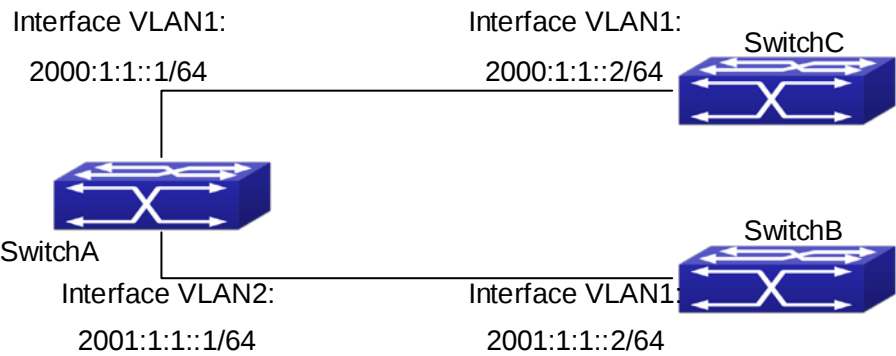


图 4-1RIPng 案例

图 4-1 为三台三层交换机组成的一个网络，三层交换机 SwitchA 三层交换机 SwitchB 和三层交换机 SwitchC 分别通过以太网口 VLAN2 和 VLAN1 相连，三台三层交换机都运行 RIPng

路由协议。设三层交换机 SwitchA（VLAN1：2001:1:1::1/64，VLAN2：2001:1:1::1/64）只与三层交换机 SwitchB（VLAN1：2001:1:1::2/64）交流三层交换机更新信息，而不向三层交换机 SwitchC（VLAN1：2001:1:1::2/64）交流三层交换机更新信息。

三层交换机 SwitchA、SwitchB、SwitchC 的配置分别如下：

a) 三层交换机 SwitchA：

启动 RIPng 协议

```
SwitchA(config)#router IPv6 rip
```

```
SwitchA(config-router)#exit
```

配置以太网口 VLAN1 的 IPv6 地址和接口运行 RIPng

```
SwitchA#config
```

```
SwitchA(config)# interface Vlan1
```

```
SwitchA(config-if-Vlan1)# IPv6 address 2000:1:1::1/64
```

```
SwitchA(config-if-Vlan1)#IPv6 router rip
```

```
SwitchA(config-if-Vlan1)#exit
```

配置以太网口 VLAN2 的 IP 地址和接口运行 RIPng

```
SwitchA(config)# interface Vlan2
```

```
SwitchA(config-if-Vlan2)# IPv6 address 2001:1:1::1/64
```

```
SwitchA(config-if-Vlan2)#IPv6 router rip
```

```
SwitchA(config-if-Vlan2)#exit
```

配置接口 VLAN1 不向交换机 SwitchC 发送 RIPng 报文

```
SwitchA(config)#
```

```
SwitchA(config-router)#passive-interface Vlan1
```

```
SwitchA(config-router)#exit
```

b) 三层交换机 SwitchB：

启动 RIP 协议

```
SwitchB(config)#router IPv6 rip
```

```
SwitchB(config-router-rip)#exit
```

配置以太网口 VLAN1 的 IP 地址和接口运行 RIPng。

```
SwitchB#config
```

```
SwitchB(config)# interface Vlan1
```

```
SwitchB(config-if)# IPv6 address 2001:1:1::2/64
```

```
SwitchB(config-if)#IPv6 router rip
```

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

c) 三层交换机 SwitchC：

启动 RIP 协议

```
SwitchC(config)#router IPv6 rip
```

```
SwitchC(config-router-rip)#exit
配置以太网口 VLAN1 的 IP 地址和接口运行 RIPng。
SwitchC#config
SwitchC(config)# interface Vlan1
SwitchC(config-if)# IPv6 address 2000:1:1::2/64
SwitchC(config-if)#IPv6 router rip
SwitchC(config-if)#exit
```

4.3.2 RIPng路由聚合功能典型案例

如下图的应用环境：

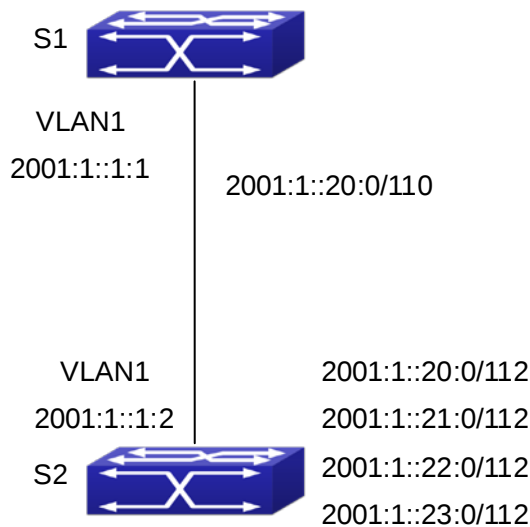


图 4-2 RIPng 路由聚合典型应用环境

在上述网络拓扑图中，S2通过接口vlan1与S1相连，S2另外有四条子网路由，分别是：2001:1::20:0/112，2001:1::21:0/112，2001:1::22:0/112，2001:1::23:0/112。S2支持路由聚合，在S2的接口vlan1上配置聚合路由2001:1::20:0/110后，通过vlan1给S1发送路由信息时，将四条子网路由聚合成一条聚合路由2001:1::20:0/110，发送给S1，而不将子网发送给邻居。减小了S1的路由表规模，节省了内存。

S1配置序列：

```
S1(config)#router ipv6 rip
S1(config)# interface Vlan1
S1(config-if-Vlan1)#IPv6 address 2001:1::1:1/112
S1(config-if-Vlan1)#IPv6 router rip
```

S2配置序列：

```
S2(config)#router ipv6 rip
S2(config)#interface vlan 1
S2(config-if-Vlan1)#IPv6 address 2001:1::1:2/112
```

```
S2(config-if-Vlan1)#IPv6 router rip
S2(Config-if-Vlan1)#ipv6 rip agg 2001:1::20:0/110
```

4.4 RIPng排错帮助

在配置、使用 RIPng 协议时，可能会由于物理连接、配置错误等原因导致 RIPng 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误和 IP Forwarding 命令开启；
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）；
- ☞ 然后，先启动 RIPng 协议（使用 router IPv6 rip 命令）并且配置接口（使用 IPv6 router rip 命令）再在相应接口配置 RIPng 协议参数；
- ☞ 接着，注意 RIPng 协议的自身特点——RIPng 三层交换机每隔 30 秒向它所有邻居三层交换机发送路由表更新信息，如果 180 秒内没有收到来自某三层交换机的信息，那么认为该三层交换机崩溃或相连的网络不可达，到该三层交换机的路由还将在路由表中保持 120 秒然后被删除。因此，如果删除某个 RIPng 路由，需等待 300 秒后才能保证 RIPng 数据库中除去此路由项。

如果仍无法解决 RIP 路由问题，那么请使用 debug IPv6 rip 调试命令，然后将 3 分钟内的 DEBUG 信息拷贝下来，发送给本公司技术服务中心。

第5章 OSPF

5.1 OSPF介绍

OSPF（Open Shortest Path First）协议即“开放最短路径优先协议”。它是一种基于链路状态的自治系统内部的动态路由协议，它通过三层交换机间交换链路状态信息来组成一个链路状态数据库，然后基于这个数据库用最短路径优先算法生成路由表。

自治系统（AS）是一个自我管理的互连网络。在大型网络中，例如Internet，极大的互连网络被分解为自治系统。连接到Internet上的大型公司网络是独立的自治系统，因为Internet上的其他主机并不由它来管理，而且它和Internet三层交换机并不共享内部路由选择信息。

每个链路状态三层交换机可提供与其邻居三层交换机的拓扑结构的信息：

- 三层交换机所连接的网段（链路）
- 连接链路的状态

链路状态信息在网络上泛洪（flooding），目的是使所有的三层交换机可以接收到第一手信息。链路状态三层交换机并不会广播包含在它们的路由表内的所有信息，相反链路状态三层交换机将发送关于已经改动的链路状态信息。链路状态三层交换机将向它们的邻居发送呼叫信息建立邻接关系，然后在邻接三层交换机之间发送链路状态通告（LSA）。邻接三层交换机将LSA复制到它们的路由选择表中，并传递这些信息到网络的剩余部分。这个过程称为泛洪（flooding）。这样，实现了向网络发送第一手信息，为网络建立更新路由的准确映射。链路状态路由选择协议使用称为代价的方法（而不是使用跳）来选择路径。代价是自动或人工赋值的。根据链路状态协议的算法，代价可以计算数据包必须穿越的跳数目、链路带宽、链路上的当前负载，甚至可以由管理员加入的权重来评价。

- 1) 当一个链路状态三层交换机进入链路状态互连网络时，它发送一个呼叫数据包（HELLO）以了解其邻居，并建立邻接关系。
- 2) 邻居用关于它们所连接的链路以及相关的代价度的信息进行应答。
- 3) 起始的三层交换机用这个信息来建立它的路由选择表。
- 4) 然后，作为定期更新的一部分。三层交换机向它的邻接三层交换机发送链路状态数据包。这个LSA包括了那个三层交换机的链路及相关代价。
- 5) 每个邻接三层交换机复制该LSA数据包，并且将LSA传递到下一个邻居（即泛洪）。
- 6) 因为三层交换机并没有在向前泛洪LSA之前重新计算路由选择数据库，因此收敛时间大大减少了。

链路状态路由选择协议的一个主要优点就是不可能形成路由无穷记数，原因是链路状态协议独立建立它们自己的路由选择信息表的方式。另一个优点是，在链路状态互连网络中收敛非常快，原因是一旦路由选择拓扑出现变动，则更新在互连网络上迅速泛洪。这些优点又

释放了三层交换机的资源，因为对不好的路由信息所花费的处理能力和带宽消耗都很少。

OSPF协议的特点：OSPF协议支持各种规模的网络，最多可以支持几百台三层交换机；路由拓扑结构变化后OSPF能够立即发送链路状态更新数据包，收敛迅速；OSPF根据收集到的链路状态采用最短路径算法计算路由，保证了无自环路由；OSPF将自治系统划分为多个域，减小了数据库尺寸、减少了占用网络带宽、降低了计算负担（根据三层交换机在自治系统所处的位置可以划分为域内三层交换机、域边界三层交换机、自治系统边界三层交换机和骨干三层交换机）；OSPF支持负载分担，支持到同一目的地址的多条等价路由；OSPF支持4级路由机制（按域内路由、域间路由、第一类外部路由和第二类外部路由顺序分级处理路由）；OSPF协议支持IP子网、支持与其他路由协议的路由的引入、支持基于接口的报文验证；OSPF支持组播发送数据包。

每个OSPF三层交换机都维护着一份描述整个自治系统拓扑结构的数据库。每一台三层交换机都搜集本地状态的信息，如可用接口信息，可达邻居信息，然后通过使用链路状态通告（即向外发送链路状态信息），本三层交换机与其他OSPF三层交换机交换链路状态信息，从而形成描述整个自治系统的链路状态数据库。根据链路状态数据库，各三层交换机构建一棵以自己为根的最短路径树，这棵树给出了到自治系统中各节点的路由。如果存在两台或两台以上的三层交换机（即多路访问网络），该网络上要选出“指定三层交换机”和“备份指定三层交换机”，指定三层交换机负责将网络的链路状态播出去，引入这一概念，有助于减少在多路访问网络上各三层交换机之间的数据流量。

OSPF协议要求将自治系统的网络划分成区域来管理，即将自治系统划分为0域（骨干域）和非0域，区域间传送的路由信息被进一步抽象概括，从而减少了整个网络的实用带宽。OSPF使用4类不同的路由，按优先顺序来说分别是区域内路由、区域间路由、第一类外部路由和第二类外部路由。区域内和区域间路由描述的是自治系统内部的网络结构，而外部路由则描述了应该如何选择到自治系统以外的目的地。第一类外部路由对应于OSPF从其它内部路由协议所引入的信息，这些路由的开销和OSPF自身路由的开销具有可比性；第二类外部路由对应于OSPF从外部路由协议所引入的信息，它们的开销远大于OSPF自身的路由开销，所以在计算路由开销时，不考虑OSPF自身的路由开销。

OSPF的区域以Backbone（骨干区域）作为核心，该区域标识为0域，所有的其他区域都必须与0域在逻辑上相连，0域必须保证连续。为此，骨干区域上特别引入了虚连接的概念以保证即使在物理上分割的该区域仍然在逻辑上具有连通性。在同一区域内的所有三层交换机的配置要保持一致。

如上所述，LSA只能在邻接三层交换机之间进行传递，OSPF协议包括五类LSA：路由器LSA、网络LSA、到其它域所在网络的汇总LSA、到自治系统边界三层交换机的汇总LSA和自治系统外部LSA。它们也可分别叫做类型1LSA、类型2LSA、类型3LSA、类型4LSA和类型5LSA。路由器LSA由OSPF域内的每个三层交换机产生，并发送给域内所有其它邻接三层交换机；网络LSA是由多路访问网络所在OSPF域的指定三层交换机产生的（为减少多路访问网络上各三层交换机之间的数据流量，多路访问网络中需要选出“指定三层交换机”和“备份指定三层交换机”，由指定三层交换机负责将网络的链路状态播出去），并发送给域内所有其它邻接三层交换机；汇总LSA由OSPF域边界三层交换机产生，并在域边界三层

交换机之间传送；自治系统外部LSA由自治系统外部边界三层交换机产生，并在整个自治系统内传递。

对于以外部链路状态通告为主的自治系统，为减少拓扑数据库的尺寸，OSPF允许将某些域配置成STUB域。类型4LSA（ASBR汇总LSA）和类型5LSA（AS外部LSA）等两种LSA不允许泛滥进入/通过STUB域。STUB域内必须使用缺省路由，STUB域的域边界三层交换机通过汇总类型3LSA来通告到STUB域的缺省路由；这些缺省路由只在STUB内泛滥，不泛滥到STUB域之外。每个STUB域对应一条缺省路由，STUB域到AS外部目的地的路由只依赖该域的缺省路由。

OSPF协议路由的简单计算过程如下：

- 每个支持 OSPF 协议的三层交换机都维护着一个描述整个自治系统拓扑结构的链路状态数据库（即 LS 数据库）。每个三层交换机根据自己周围的网络拓扑结构生成一条链路状态通告（即路由器 LSA），通过相互之间发送链路状态更新包（LSU）将各自的 LSA 发送给网络中的其它三层交换机。这样，每台三层交换机都收到了其它三层交换机的 LSA，所有 LSA 在一起便形成了链路状态数据库。
- 由于一条 LSA 是对该三层交换机周围网络拓扑结构的描述，那么 LS 数据库则是对整个网络的拓扑结构的描述。三层交换机很容易根据 LS 数据库建立一张带权有向图，显然自治系统内的所有三层交换机都将得到一张相同的网络拓扑图。
- 每台三层交换机都使用最短路径 SPF 算法计算出一棵以自己为根的最短路径树，该树给出了到自治系统中各个节点的路由，外部路由信息为叶子节点，外部路由可根据广播它的三层交换机进行标记以记录关于自治系统的额外信息。由此可见，各个三层交换机得到的路由表是不同的。

OSPF 协议是 IETF 组织开发的，目前广泛使用的是 OSPFv2 版本是参照 RFC2328 中描述的内容实现的。

5.2 OSPF配置

三层交换机的 OSPF 配置有自身的特点，简单说分两步：1、在全局使能 OSPF；2、配置接口所属的区域。配置任务序列如下：

1. 启动 OSPF 协议（必须）
 - （1）启动/关闭 OSPF 协议（必须）
 - （2）配置运行 OSPF 三层交换机的 ID 号（可选）
 - （3）配置运行 OSPF 的网络范围（可选）
 - （4）配置接口所属的域（必须）
2. 配置 OSPF 辅助参数（可选）
 - （1）配置 OSPF 发包机制参数
 - 1）配置 OSPF 数据包的验证
 - 2）配置 OSPF 接口为只收不发
 - 3）配置接口发送数据包的代价

- 4) 配置 OSPF 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）
- (2) 配置 OSPF 引入路由参数
 - 1) 配置引入外部路由的缺省参数（缺省类型、缺省标记值、缺省代价值）
 - 2) 配置在 OSPF 中引入其它协议的路由
- (3) 配置 OSPF 引入其他 OSPF 进程路由功能
 - 1) 启动 OSPF 引入其他 OSPF 进程路由功能
 - 2) 显示相关配置信息
 - 3) 调试
- (4) 配置 OSPF 协议其它参数
 - 1) 配置 OSPF 路由协议优先级
 - 2) 配置 OSPF STUB 域及缺省路由的代价
 - 3) 配置 OSPF 虚链路
 - 4) 配置接口在选举指定三层交换机 DR 中的优先级
 - 5) 配置打开/关闭记录 OSPF 邻居状态变化日志功能
 - 6) 过滤 OSPF 计算得到的路由
3. 关闭 OSPF 协议

1. 启动 OSPF 协议

在三层交换机上运行 OSPF 路由协议的基本配置也很简单，通常只需打开 OSPF 开关、配置接口所在 OSPF 域即可，OSPF 协议的参数都使用缺省值。如需修改 OSPF 协议参数值，参看 2.配置 OSPF 辅助参数。

命令	解释
全局配置模式	
[no] router ospf [process <id>] [VRF Name]	启动 OSPF 协议进程，本命令的 no 操作关闭 OSPF 协议。（必须）
OSPF 协议配置模式	
router-id <router_id> no router-id	配置运行 OSPF 协议三层交换机的 ID 号；本命令的 no 操作取消三层交换机的 ID 号。缺省为选择某个接口的 IP 地址作为三层交换机 ID。（可选）
[no] network {<network> <mask> <network>/<prefix>} area <area_id>	配置某个网段属于某个区域，本命令的 no 操作为取消该配置。（必须）

2. 配置 OSPF 辅助参数

(1) 配置 OSPF 发包机制参数

- 1) 配置 OSPF 数据包的验证
- 2) 配置 OSPF 接口为只收不发
- 3) 配置接口发送数据包的代价

命令	解释
接口配置模式	
ip ospf authentication { message-digest null } no ip ospf authentication	配置接口接受 OSPF 数据包所需要的验证方式 本命令的 no 操作恢复为缺省值。
ip ospf authentication-key LINE no ip ospf authentication-key	配置接口接受 OSPF 数据包所需要的验证方式的 key；本命令的 no 操作恢复为缺省值。
[no] passive-interface <ifname> [<ip-address>]	配置在特定接口上不发送 hello 分组，本命令的 no 操作取消该功能。
ip ospf cost <cost > no ip ospf cost	指定接口运行 OSPF 协议所需的代价；本命令的 no 操作恢复代价的缺省值。

- 4) 配置 OSPF 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）

命令	解释
接口配置模式	
ip ospf hello-interval <time> no ip ospf hello-interval	配置接口周期发送 HELLO 数据包的时间间隔； 本命令的 no 操作恢复为缺省值。
ip ospf dead-interval <time > no ip ospf dead-interval	配置认定相邻三层交换机失效的时间间隔；本命令的 no 操作恢复为缺省值。
ip ospf transit-delay <time> no ip ospf transit-delay	设置在接口上传送链路状态广播的时延值；本命令的 no 操作恢复为缺省值。
ip ospf retransmit <time> no ip ospf retransmit	配置接口与邻接三层交换机之间传送链路状态通告时的重传间隔；本命令的 no 操作恢复为缺省值。

(2) 配置 OSPF 引入路由参数

配置在 OSPF 中引入其它协议的路由

命令	解释
OSPF 协议配置模式	

redistribute { bgp connected static rip kernel } [metric-type { 1 2 }] [tag <tag>] [metric <cost_value>] [router-map <WORD>] no redistribute { bgp connected static rip kernel }	引入其他协议发现路由和静态路由作为外部路由信息；本命令的 no 操作取消引入的外部路由信息。
---	--

(3) 配置 OSPF 引入其他 OSPF 进程路由功能

1) 启动 OSPF 引入其他 OSPF 进程路由功能

命令	解释
Router ospf 配置模式	
redistribute ospf [<process-id> [metric<value>] [metric-type {1 2}][route-map<word>] no redistribute ospf [<process-id> [metric<value>] [metric-type {1 2}][route-map<word>]	启动或者关闭 OSPF 引入其他 OSPF 进程路由功能。

2) 显示相关配置信息

命令	解释
特权模式或者配置模式	
show ip ospf [<process-id> redistribute	显示 ospf 进程引入其它外部路由的配置信息。

3) 调试

命令	解释
特权模式	
debug ospf redistribute message send no debug ospf redistribute message send debug ospf redistribute route receive no debug ospf redistribute route receive	打开或关闭 ospf 进程引入其他 ospf 进程路由的命令下发调试开关。 打开或关闭 ospf 进程收到 nsm 发来的路由信息的调试开关。

(4) 配置 OSPF 协议其它参数

1) 配置 OSPF spf 算法时间的计算

2) 配置 OSPF 链路状态数据库中 LSA 的数目限制

3) 配置 OSPF 各种区域参数

命令	解释
OSPF 协议配置模式	
timers spf <interval> no timers spf	配置 OSPF 的 SPF 定时器；本命令的 no 操作恢复为缺省值。
overflow database {<max-LSA> [hard soft] external <max-LSA> <recover time>} no overflow database [external <max-LSA> <recover time>]	配置当前 OSPF 进程数据库中 LSA 的限制；本命令的 no 操作恢复为缺省值。
area <id> {authentication [message-digest] default-cost <cost> filter-list {access prefix} <WORD> {in out} nssa [default-information-originate no-redistribution no-summary translator-role] range <range> stub [no-summary] virtual-link <neighbor>} no area <id> {authentication default-cost filter-list {access prefix} <WORD> {in out} nssa [default-information-originate no-redistribution no-summary translator-role] range <range> stub [no-summary] virtual-link <neighbor>}	配置 OSPF 区域下的参数（STUB 域，NSSA 域和虚链路）；本命令的 no 操作恢复为缺省值。

4) 配置接口在选举指定三层交换机 DR 中的优先级

命令	解释
接口配置模式	
ip ospf priority <priority> no ip ospf priority	配置接口在选举“指定三层交换机”时的优先级；本命令的 no 操作为恢复优先级缺省值。

5) 配置打开/关闭记录 OSPF 邻居状态变化日志功能

命令	解释
OSPF 协议配置模式	
log-adjacency-changes detail no log-adjacency-changes detail	设置是否记录 ospf 邻居变化的日志信息。

6) 过滤 OSPF 计算得到的路由

命令	解释
OSPF 协议配置模式	
filter-policy <access-list-name> no filter-policy	使用访问列表对 OSPF 计算得到的路由进行过滤。no 命令取消路由过滤。

3. 关闭 OSPF 协议

命令	解释
全局配置模式	
no router ospf [process <id>]	关闭 OSPF 路由协议。

5.3 OSPF案例

5.3.1 OSPF典型案例

案例一：OSPF自治系统

以五台三层交换机为例组成一个OSPF自治系统，OSPF区域划分见下图。

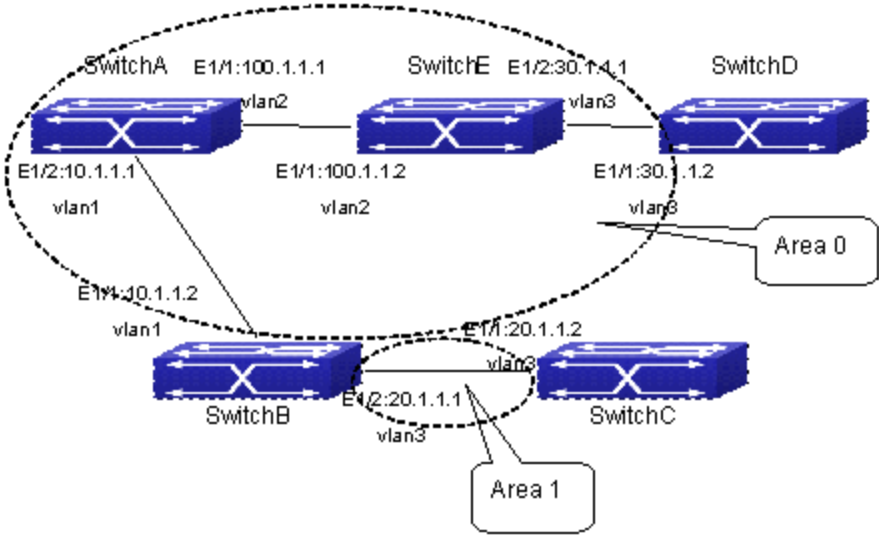


图 5-1 OSPF自治系统网络拓扑示意图

三层交换机Switch1- Switch5的配置分别如下：

三层交换机Switch1：

配置接口vlan1的IP地址。

Switch1#config

Switch1(config)# interface vlan 1

Switch1(config-if-vlan1)# ip address 10.1.1.1 255.255.255.0

```
Switch1(config-if-vlan1)#exit
```

配置接口vlan2的IP地址

```
Switch1(config)# interface vlan 2
```

```
Switch1(config-if-vlan2)# ip address 100.1.1.1 255.255.255.0
```

```
Switch1 (config-if-vlan2)#exit
```

启动OSPF协议，配置接口vlan1、vlan2所属域号

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 10.1.1.0/24 area 0
```

```
Switch1(config-router)#network 100.1.1.0/24 area 0
```

```
Switch1(config-router)#exit
```

```
Switch1(config)#exit
```

```
Switch1#
```

三层交换机Switch2:

配置接口vlan1和vlan2的IP地址。

```
Switch2#config
```

```
Switch2(config)# interface vlan 1
```

```
Switch2(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
```

```
Switch2(config-if-vlan1)#no shutdown
```

```
Switch2(config-if-vlan1)#exit
```

```
Switch2(config)# interface vlan 3
```

```
Switch2(config-if-vlan3)# ip address 20.1.1.1 255.255.255.0
```

```
Switch2(config-if-vlan3)#no shutdown
```

```
Switch2(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan1和vlan3所属OSPF区域

```
Switch2(config)#router ospf
```

```
Switch2(config-router)# network 10.1.1.0/24 area 0
```

```
Switch2(config-router)# network 20.1.1.0/24 area 1
```

```
Switch2(config-router)#exit
```

```
Switch2(config)#exit
```

```
Switch2#
```

三层交换机Switch3:

配置接口vlan3的IP地址。

```
Switch3#config
```

```
Switch3(config)# interface vlan 3
```

```
Switch3(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
```

```
Switch3(config-if-vlan3)#no shutdown
```

```
Switch3(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan3所属OSPF区域

```
Switch3(config)#router ospf
Switch3(config-router)# network 20.1.1.0/24 area 1
Switch3(config-router)#exit
Switch3(config)#exit
Switch3#
```

三层交换机Switch4:
配置接口vlan3的IP地址。

```
Switch4#config
Switch4(config)# interface vlan 3
Switch4(config-if-vlan3)# ip address 30.1.1.2 255.255.255.0
Switch4(config-if-vlan3)#no shutdown
Switch4(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan3所属OSPF区域

```
Switch4(config)#router ospf
Switch4(config-router)# network 30.1.1.0/24 area 0
Switch4(config-router)#exit
Switch4(config)#exit
Switch4#
```

三层交换机Switch5:
配置接口vlan2的IP地址。

```
Switch5#config
Switch5(config)# interface vlan 2
Switch5(config-if-vlan2)# ip address 100.1.1.2 255.255.255.0
Switch5(config-if-vlan2)#no shutdown
Switch5(config-if-vlan2)#exit
```

配置接口vlan3的IP地址

```
Switch5(config)# interface vlan 3
Switch5(config-if-vlan3)# ip address 30.1.1.1 255.255.255.0
Switch5(config-if-vlan3)#no shutdown
Switch5(config-if-vlan3)#exit
```

启动OSPF协议，配置接口vlan2和vlan3所属域号

```
Switch5(config)#router ospf
Switch5(config-router)# network 30.1.1.0/24 area 0
Switch5(config-router)# network 100.1.1.0/24 area 0
Switch5(config-router)#exit
Switch5(config)#exit
Switch5#
```

案例二：OSPF协议典型复杂拓扑

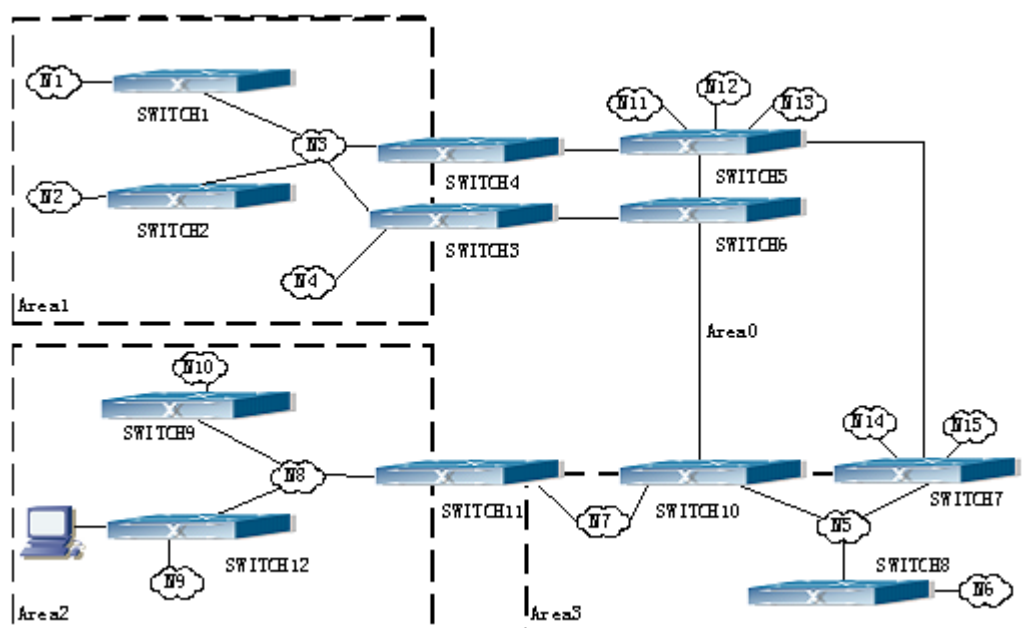


图 5-2 复杂典型 OSPF 自治系统

图 5-2 为一个典型复杂的 OSPF 自治系统网络拓扑。域 1 包括网络 N1-N4 和三层交换机 Switch1-Switch4，域 2 包括网络 N8-N10、主机 H1 与三层交换机 Switch9，域 3 包括网络 N5-N7 和三层交换机 Switch7、Switch8、Switch10 和 Switch11，且配置成网络 N8-N10 和主机 H1 使用一条汇总路由（即将域 3 定义为 STUB 域）。三层交换机 Switch1、Switch2、Switch5、Switch6、Switch8、Switch9 和 Switch12 是域内三层交换机，三层交换机 Switch3、Switch4、Switch7、Switch10 和 Switch11 是域边界三层交换机，三层交换机 Switch5 和 Switch7 是自治系统边界三层交换机。

就域 1 而言，三层交换机 Switch1 和 Switch2 是域内三层交换机，域边界三层交换机 Switch3 和 Switch4 负责向域 1 通告所有到该域外目的地的距离代价，同时 Switch3 和 Switch4 还必须向域 1 通告自治系统边界三层交换机 Switch5 和 Switch7 的位置，来自 Switch5 和 Switch7 的自治系统外部链路状态通告在整个自治系统内泛滥。当自治系统外部链路状态通告在域 1 内泛滥时，这些 LSA 被包含在域 1 数据库内，从而得到到网络 N11-N15 的路由。

另外，三层交换机 Switch3 和 Switch4 也必须向骨干域（即 0 域，所有非 0 域必须通过 0 域相连，不允许它们直接相连）汇总域 1 的拓扑结构，通告域 1 所包含的网络（即 N1-N4）以及从 Switch3 和 Switch4 到这些网络的代价。由于骨干域要求必须是连通的，所以在骨干三层交换机 Switch10 和 Switch11 之间必须建立一条虚链接。域边界三层交换机之间通过骨干三层交换机来交换汇总信息，每一个域边界三层交换机监听来自其它域边界三层交换机的汇总信息。

虚链路不仅用于保证骨干域的连通性，还用于健壮骨干域。比如，骨干三层交换机 Switch6 和 Switch10 之间的链路如果被切断，这将导致骨干域不连续。通过在骨干三层交换

机Switch7和Switch10之间建立一条虚链路，骨干域会变得更健壮一些。同时，Switch7和Switch10之间的虚链接提供了到域3和三层交换机Switch7的一条更短的路径。

以域1为例，假设三层交换机Switch1的接口Vlan2的IP地址为10.1.1.1；三层交换机Switch2的接口VLAN2的IP地址为10.1.1.2；三层交换机Switch3的接口VLAN2的IP地址为10.1.1.3；三层交换机Switch4的接口VLAN2的IP地址为10.1.1.4。Switch1使用以太网口VLAN1与网络N1相连，IP地址为20.1.1.1；Switch2使用以太网口VLAN1与网络N2相连，IP地址为20.1.2.1；Switch3使用以太网口VLAN3与网络N4相连，IP地址为20.1.3.1；都属于域1。Switch3使用以太网口VLAN1与三层交换机Switch6相连，IP地址为10.1.5.1；Switch4使用以太网口VLAN1与三层交换机Switch5相连，IP地址为10.1.6.1；都属于域0。域1内的三层交换机之间采用简单密钥认证，域1边界三层交换机与域0骨干三层交换机之间采用MD5密钥认证。

下面仅给出域1内各三层交换机的配置，其他域三层交换机的配置省略。Switch1、Switch2、Switch3和Switch4的配置分别如下：

1)Switch1:

配置接口vlan2的IP地址

```
Switch1#config
```

```
Switch1(config)# interface vlan 2
```

```
Switch1(config-if-Vlan2)# ip address 10.1.1.1 255.255.255.0
```

```
Switch1(config-if-Vlan2)#exit
```

启动OSPF协议，配置接口vlan2所属域号

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 10.1.1.0/24 area 1
```

```
Switch1(config-router)#exit
```

配置简单密钥认证

```
Switch1(config)#interface vlan 2
```

```
Switch1(config-if-Vlan2)#ip ospf authentication
```

```
Switch1(config-if-Vlan2)#ip ospf authentication-key DCS
```

```
Switch1(config-if-Vlan2)#exit
```

配置接口vlan1的IP地址，配置所属域号

```
Switch1(config)# interface vlan 1
```

```
Switch1(config-if-Vlan1)#ip address 20.1.1.1 255.255.255.0
```

```
Switch1(config-if-Vlan1)#exit
```

```
Switch1(config)#router ospf
```

```
Switch1(config-router)#network 20.1.1.0/24 area 1
```

```
Switch1(config-router)#exit
```

2)Switch2:

配置接口vlan2的IP地址

```
Switch2#config
```

```
Switch2(config)# interface vlan 2
```

```
Switch2(config-If-Vlan2)# ip address 10.1.1.2 255.255.255.0
```

```
Switch2(config-If-Vlan2)#exit
```

启动OSPF协议，配置接口vlan2所属域号

```
Switch2(config)#router ospf
```

```
Switch2(config-router)#network 10.1.1.0/24 area 1
```

```
Switch2(config-router)#exit
```

```
Switch2(config)#interface vlan 2
```

配置简单密钥认证

```
Switch2(config)#interface vlan 2
```

```
Switch2(config-If-Vlan2)#ip ospf authentication
```

```
Switch2(config-If-Vlan2)#ip ospf authentication-key DCS
```

```
Switch2(config-If-Vlan2)#exit
```

配置接口vlan1的IP地址，配置所属域号

```
Switch2(config)# interface vlan 1
```

```
Switch2(config-If-Vlan1)#ip address 20.1.2.1 255.255.255.0
```

```
Switch2(config-If-Vlan1)#exit
```

```
Switch2(config)#router ospf
```

```
Switch2(config-router)#network 20.1.2.0/24 area 1
```

```
Switch2(config-router)#exit
```

```
Switch2(config)#exit
```

```
Switch2#
```

3)Switch3:

配置接口vlan2的IP地址

```
Switch3#config
```

```
Switch3(config)# interface vlan 2
```

```
Switch3(config-If-Vlan2)# ip address 10.1.1.3 255.255.255.0
```

```
Switch3(config-If-Vlan2)#exit
```

启动OSPF协议，配置接口vlan2所属域号

```
Switch3(config)#router ospf
```

```
Switch3(config-router)#network 10.1.1.0/24 area 1
```

```
Switch3(config-router)#exit
```

配置简单密钥认证

```
Switch3(config)#interface vlan 2
```

```
Switch3(config-If-Vlan2)#ip ospf authentication
```

```
Switch3(config-If-Vlan2)#ip ospf authentication-key DCS
```

```
Switch3(config-If-Vlan2)#exit
```

配置接口vlan3的IP地址，配置所属域号

```
Switch3(config)# interface vlan 3
```

```
Switch3(config-If-Vlan3)#ip address 20.1.3.1 255.255.255.0
```

```
Switch3(config-If-Vlan3)#exit
```

```
Switch3(config)#router ospf
```

```
Switch3(config-router)#network 20.1.3.0/24 area 1
```

```
Switch3(config-router)#exit
```

配置接口vlan1的IP地址，并配置接口所属域号

```
Switch3(config)# interface vlan 1
```

```
Switch3(config-If-Vlan1)#ip address 10.1.5.1 255.255.255.0
```

```
Switch3(config-If-Vlan1)#exit
```

```
Switch3(config)#router ospf
```

```
Switch3(config-router)#network 10.1.5.0/24 area 0
```

```
Switch3(config-router)#exit
```

配置MD5密钥认证

```
Switch3(config)#interface vlan 1
```

```
Switch3 (config-If-Vlan1)#ip ospf authentication message-digest
```

```
Switch3 (config-If-Vlan1)#ip ospf authentication-key DCS
```

```
Switch3 (config-If-Vlan1)#exit
```

```
Switch3(config)#exit
```

```
Switch3#
```

4)Switch4:

配置接口vlan2的IP地址

```
Switch4#config
```

```
Switch4(config)# interface vlan 2
```

```
Switch4(config-If-Vlan2)# ip address 10.1.1.4 255.255.255.0
```

```
Switch4(config-If-Vlan2)#exit
```

启动OSPF协议，配置接口vlan2所属域号

```
Switch4(config)#router ospf
```

```
Switch4(config-router)#network 10.1.1.0/24 area 1
```

```
Switch4(config-router)#exit
```

配置简单密钥认证

```
Switch4(config)#interface vlan 2
```

```
Switch4(config-If-Vlan2)#ip ospf authentication
```

```
Switch4(config-If-Vlan2)#ip ospf authentication-key DCS
```

```
Switch4(config-If-Vlan2)#exit
```

配置接口vlan1的IP地址，并配置接口所属域号

```
Switch4(config)# interface vlan 1
```

```
Switch4(config-If-Vlan1)# ip address 10.1.6.1 255.255.255.0
```

```
Switch4(config-If-Vlan1)#exit
```

```
Switch4(config)#router ospf
Switch4(config-router)#network 10.1.6.0/24 area 0
Switch4(config-router)#exit
配置MD5密钥认证
Switch4(config)#interface vlan 1
Switch4(config-if-Vlan1)#ip ospf authentication message-digest
Switch4(config-if-Vlan1)#ip ospf authentication-key DCS
Switch4(config-if-Vlan1)#exit
Switch4(config)#exit
Switch4#
```

案例三：OSPF引入其他OSPF进程路由功能

如下图所示，一台运行 OSPF 路由协议的交换机连接两个网络，网络 A 与网络 B，出于某种原因，要求网络 A 可以学到网络 B 的路由，但网络 B 不可以学到网络 A 的路由，根据需求，可配置在 interface vlan1 与 interface vlan2 上分别起 OSPF 的两个进程，同时配置 interface vlan1 所属的 OSPF 进程引入 interface vlan2 所属的 OSPF 进程的路由，但不能配置 interface vlan2 所属的 OSPF 进程引入 interface vlan1 所属的 OSPF 进程的路由。

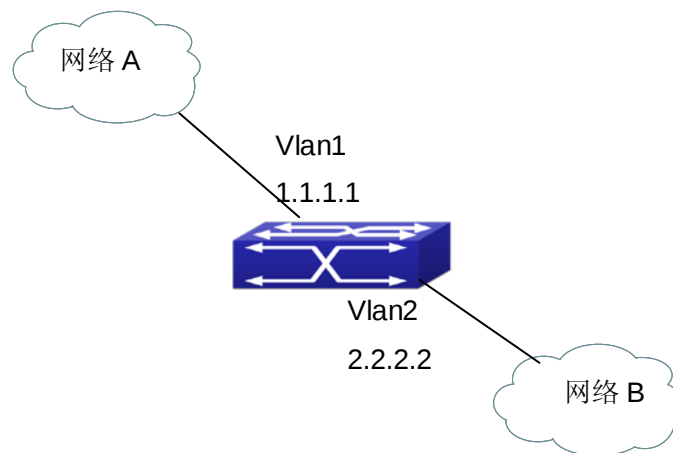


图 5-3 OSPF 引入其他 OSPF 进程路由功能案例

我们可以进行如下配置：

```
Switch(config)#interface vlan 1
Switch(Config-if-Vlan1)#ip address 1.1.1.1 255.255.255.0
Switch(Config-if-Vlan1)#exit
Switch(config)#interface vlan 2
Switch(Config-if-Vlan2)#ip address 2.2.2.2 255.255.255.0
Switch(Config-if-Vlan2)#exit
```

```

Switch(config)#router ospf 10
Switch(config-router)#network 2.2.2.0/24 area 1
Switch(config-router)#exit
Switch(config)#router ospf 20
Switch(config-router)#network 1.1.1.0/24 area 1
Switch(config-router)#redistribute ospf 10
Switch(config-router)#exit

```

5.3.2 OSPF VPN 典型案例

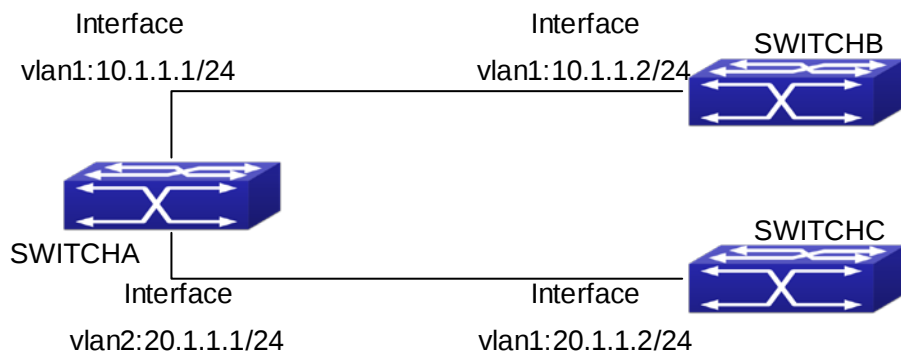


图 5-4 OSPF VPN 案例

图 5-4 为三台三层交换机组成的网络。交换机 SwitchA 充当 PE，交换机 SwitchB 和交换机 SwitchC 充当 CE1 与 CE2。PE 通过接口 vlan1 与 vlan2 分别与 CE1 和 CE2 相连。PE 与 CE 之间通过 OSPF 协议交换路由信息

a) 作为 PE 的三层交换机 SwitchA:

配置 VPN 路由/转发实例 vpnb 和 vpnc

```
SwitchA#config
```

```
SwitchA(config)#ip vrf vpnb
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

```
SwitchA#(config)
```

```
SwitchA(config)#ip vrf vpnc
```

```
SwitchA(config-vrf)#
```

```
SwitchA(config-vrf)#exit
```

配置接口 vlan1 与 vlan2 分别与 vpnb 和 vpnc 相关联，并且配置 IP 地址

```
SwitchA(config)#in vlan1
```

```
SwitchA(config-if-Vlan1)#ip vrf forwarding vpnb
```

```
SwitchA(config-if-Vlan1)#ip address 10.1.1.1 255.255.255.0
```

```
SwitchA(config-if-Vlan1)#exit
```

```
SwitchA(config)#in vlan2
SwitchA(config-if-Vlan2)#ip vrf forwarding vpnc
SwitchA(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
SwitchA(config-if-Vlan2)#exit
```

分别配置与 vpnb 和 vpnc 相关联的 OSPF 实例

```
SwitchA(config)#
SwitchA(config)#router ospf 100 vpnb
SwitchA(config-router)#network 10.1.1.0/24 area 0
SwitchA(config-router)#redistribute bgp
SwitchA(config-router)#exit
SwitchA(config)#router ospf 200 vpnc
SwitchA(config-router)#network 20.1.1.0/24 area 0
SwitchA(config-router)#redistribute bgp
```

b) CE1 三层交换机 SwitchB:

配置以太网口 E1/2 的 IP 地址。

```
SwitchB#config
SwitchB(config)# interface Vlan1
SwitchB(config-if-vlan1)# ip address 10.1.1.2 255.255.255.0
SwitchB (config-if-vlan1)#exit
```

启动 OSPF 协议，配置 OSPF 网段

```
SwitchB(config)#router ospf
SwitchB(config-router-rip)#network 10.1.1.0/24 area 0
SwitchB(config-router-rip)#exit
```

c) CE2 三层交换机 SwitchC:

配置以太网口 E1/2 的 IP 地址。

```
SwitchC#config
SwitchC(config)# interface Vlan1
SwitchC(config-if-vlan1)# ip address 20.1.1.2 255.255.255.0
SwitchC (config-if-vlan1)#exit
```

启动 OSPF 协议，配置 OSPF 网段

```
SwitchC(config)#router ospf
SwitchC(config-router)#network 20.1.1.0/24 area 0
SwitchC(config-router)#exit
```

5.4 OSPF排错帮助

在配置、使用 OSPF 协议时，可能会由于物理连接、配置错误等原因导致 OSPF 协议

未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 然后在各接口上配置不同网段的 IP 地址；
- ☞ 然后，先启动 OSPF 协议（使用 `router ospf` 命令）再在相应接口配置所属 OSPF 域；
- ☞ 接着，注意 OSPF 协议的自身特点——OSPF 骨干域（0 域）必须保证是连续的；如果不连续，要使用虚链路（`virtual link`）来保证；所有非 0 域只能通过 0 域与其他非 0 域相连，不允许非 0 域直接相连；边界三层交换机是指该三层交换机的一部分接口属于 0 域，而另外一部分接口属于非 0 域；对于广播网等多访问网，需要选举指定三层交换机 DR。

第6章 OSPFv3

6.1 OSPFv3 介绍

OSPFv3 (Open Shortest Path First) 协议即“开放最短路径优先协议”第三版，是OSPF协议的IPv6版本。它是一种基于链路状态的自治系统内部的动态路由协议，它通过路由器或三层交换机之间交换链路状态信息来组成一个链路状态数据库，然后基于这个数据库用最短路径优先算法生成路由表。

自治系统 (AS) 是一个自我管理的互连网络。在大型网络中，例如Internet，极大的互连网络被分解为自治系统。连接到Internet上的大型公司网络是独立的自治系统，因为Internet上的其他主机并不由它来管理，而且它和Internet三层交换机并不共享内部路由选择信息。

每个链路状态三层交换机可提供与其邻居三层交换机的拓扑结构的信息：

- 三层交换机所连接的网段（链路）
- 连接链路的状态

链路状态信息在网络上泛洪 (flooding)，目的是使所有的三层交换机可以接收到第一手信息。链路状态三层交换机并不会广播包含在它们的路由表内的所有信息，相反链路状态三层交换机将发送关于已经改动的链路状态信息。链路状态三层交换机将向它们的邻居发送呼叫信息建立邻接关系，然后在邻接三层交换机之间发送链路状态通告 (LSA)。邻接三层交换机将LSA复制到它们的路由选择表中，并传递这些信息到网络的剩余部分。这个过程称为泛洪 (flooding)。这样，实现了向网络发送第一手信息，为网络建立更新路由的准确映射。链路状态路由选择协议使用称为代价的方法（而不是使用跳）来选择路径。代价是自动或人工赋值的。根据链路状态协议的算法，代价可以计算数据包必须穿越的跳数目、链路带宽、链路上的当前负载，甚至可以由管理员加入的权重来评价。

- 1) 当一个链路状态三层交换机进入链路状态互连网络时，它发送一个呼叫数据包 (HELLO) 以了解其邻居，并建立邻接关系。
- 2) 邻居用关于它们所连接的链路以及相关的代价度的信息进行应答。
- 3) 起始的三层交换机用这个信息来建立它的路由选择表。
- 4) 然后，作为定期更新的一部分。三层交换机向它的邻接三层交换机发送链路状态数据包。这个LSA包括了那个三层交换机的链路及相关代价。
- 5) 每个邻接三层交换机复制该LSA数据包，并且将LSA传递到下一个邻居（即泛洪）。
- 6) 因为三层交换机并没有在向前泛洪LSA之前重新计算路由选择数据库，因此收敛时间大大减少了。

链路状态路由选择协议的一个主要优点就是路由无穷记数不可能形成，原因是链路状态协议的路由器独立建立它们自己的路由选择信息表。另一个优点是，在链路状态互连网络中

收敛非常快，原因是一旦路由选择拓扑出现变动，则更新在互连网络上迅速泛洪。这些优点又释放了三层交换机的资源，因为对不好的路由信息所花费的处理能力和带宽消耗都很少。

OSPFv3协议的特点：OSPFv3协议支持各种规模的网络，最多可以支持几百台三层交换机；路由拓扑结构变化后OSPFv3能够立即发送链路状态更新数据包，收敛迅速；OSPFv3根据收集到的链路状态采用最短路径算法计算路由，保证了无自环路由；OSPFv3将自治系统划分为多个域，减小了数据库尺寸、减少了占用网络带宽、降低了计算负担（根据三层交换机在自治系统中所处的位置可以划分为域内三层交换机、域边界三层交换机、自治系统边界三层交换机和骨干三层交换机）；OSPFv3支持负载分担，支持到同一目的地址的多条等价路由；OSPFv3支持4级路由机制（按域内路由、域间路由、第一类外部路由和第二类外部路由顺序分级处理路由）；OSPFv3协议支持IP子网、支持其他路由协议的路由的引入；OSPFv3支持组播发送数据包。

每个OSPFv3三层交换机都维护着一份描述整个自治系统拓扑结构的数据库。每一台三层交换机都搜集本地状态的信息，如可用接口信息，可达邻居信息，然后通过使用链路状态通告（即向外发送链路状态信息），本三层交换机与其他OSPFv3三层交换机交换链路状态信息，从而形成描述整个自治系统的链路状态数据库。根据链路状态数据库，各三层交换机构建一棵以自己为根的最短路径树，这棵树给出了到自治系统中各节点的路由。如果存在两台或两台以上的三层交换机（即多路访问网络），该网络上要选出“指定三层交换机”和“备份指定三层交换机”，指定三层交换机负责将网络的链路状态播出去，引入这一概念，有助于减少在多路访问网络上各三层交换机之间的数据流量。

OSPFv3协议要求将自治系统的网络划分成区域来管理，即将自治系统划分为0域（骨干域）和非0域，区域间传送的路由信息被进一步抽象概括，从而减少了整个网络的实用带宽。OSPFv3使用4类不同的路由，按优先顺序来说分别是区域内路由、区域间路由、第一类外部路由和第二类外部路由。区域内和区域间路由描述的是自治系统内部的网络结构，而外部路由则描述了应该如何选择到自治系统以外的目的地。第一类外部路由对应于OSPFv3从其它内部路由协议所引入的信息，这些路由的开销和OSPFv3自身路由的开销具有可比性；第二类外部路由对应于OSPFv3从外部路由协议所引入的信息，它们的开销远大于OSPFv3自身的路由开销，所以在计算路由开销时，不考虑OSPFv3自身的路由开销。

OSPFv3的区域以Backbone（骨干区域）作为核心，该区域标识为0域，所有的其他区域都必须与0域在逻辑上相连，0域必须保证连续。为此，骨干区域上特别引入了虚连接的概念以保证即使在物理上分割的该区域仍然在逻辑上具有连通性。在同一区域内的所有三层交换机的配置要保持一致。

如上所述，LSA只能在邻接三层交换机之间进行传递，OSPFv3协议包括七类LSA：链路LSA、域内前缀LSA、路由器LSA、网络LSA、域间前缀LSA、域间路由器LSA和自治系统外部LSA。路由器LSA由OSPFv3域内的每个三层交换机产生，并发送给域内所有其它邻接三层交换机；网络LSA是由多路访问网络所在OSPFv3域的指定三层交换机产生的（为减少多路访问网络上各三层交换机之间的数据流量，多路访问网络中需要选出“指定三层交换机”和“备份指定三层交换机”，由指定三层交换机负责将网络的链路状态播出去），并发送给域内所有其它邻接三层交换机；域间前缀LSA和域间路由器LSA由OSPFv3域边界三层交换

机产生，并在域边界三层交换机之间传送；自治系统外部LSA由自治系统外部边界三层交换机产生，并在整个自治系统内传递。链路LSA由链路上的三层交换机产生，并且发送给链路上的其它三层交换机。域内前缀LSA由域内每个链路上的指定路由器产生，并且泛洪到整个域内。

对于以外部链路状态通告为主的自治系统，为减少拓扑数据库的尺寸，OSPFv3允许将某些域配置成STUB域。路由器LSA、网络LSA、域间前缀LSA、链路LSA、域内前缀LSA允许被通告到STUB域。STUB域内必须使用缺省路由，STUB域的域边界三层交换机通过域间前缀LSA来通告到STUB域的缺省路由；这些缺省路由只在STUB内泛滥，不泛滥到STUB域之外。每个STUB域对应一条缺省路由，STUB域到AS外部目的地的路由只依赖该域的缺省路由。

OSPFv3协议路由的简单计算过程如下：

- 1) 每个支持 OSPFv3 协议的三层交换机都维护着一个描述整个自治系统拓扑结构的链路状态数据库（即 LS 数据库）。每个三层交换机根据自己周围的网络拓扑结构生成一条链路状态通告（即路由器 LSA），通过相互之间发送链路状态更新包（LSU）将各自的 LSA 发送给网络中的其它三层交换机。这样，每台三层交换机都收到了其它三层交换机的 LSA，所有 LSA 在一起便形成了链路状态数据库。
- 2) 由于一条 LSA 是对该三层交换机周围网络拓扑结构的描述，那么 LS 数据库则是对整个网络的拓扑结构的描述。三层交换机很容易根据 LS 数据库建立一张带权有向图，显然自治系统内的所有三层交换机都将得到一张相同的网络拓扑图。
- 3) 每台三层交换机都使用最短路径优先 SPF 算法计算出一棵以自己为根的最短路径树，该树给出了到自治系统中各个节点的路由，外部路由信息为叶子节点，外部路由可根据广播它的三层交换机进行标记以记录关于自治系统的额外信息。由此可见，各个三层交换机得到的路由表是不同的。

OSPFv3协议是IETF组织开发的，目前使用的是OSPFv3版本是参照RFC2328和RFC2740中描述的内容实现的。

由于IPv6网络还在发展中，有时网络中存在不支持IPv6的网络环境，这就需要通过隧道进行IPv6的操作，所以我们的OSPFv3支持在配置隧道上的配置和运行，以IPv4单播的方式穿过不支持IPv6的网络。

6.2 OSPFv3 配置

OSPFv3 配置任务序列：

1. 启动 OSPFv3 协议（必须）
 - (1) 启动/关闭 OSPFv3 协议（必须）
 - (2) 配置运行 OSPFv3 的三层交换机的 router-id
 - (3) 配置运行 OSPFv3 的网络范围（可选）
 - (4) 在接口使能 OSPFv3（必须）

2. 配置 OSPFv3 辅助参数（可选）

- (1) 配置 OSPFv3 发包机制参数
 - 1) 配置 OSPFv3 接口为只收不发
 - 2) 配置接口发送数据包的代价
 - 3) 配置 OSPFv3 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）
- (2) 配置 OSPFv3 引入路由参数
 - 1) 配置引入外部路由的缺省参数（缺省类型、缺省标记值、缺省代价值）
 - 2) 配置在 OSPFv3 中引入其它协议的路由
- (3) 配置 OSPFv3 引入其他 OSPFv3 进程路由功能
 - 1) 启动 OSPFv3 引入其他 OSPFv3 进程路由功能
 - 2) 显示相关配置信息
 - 3) 调试
- (4) 配置 OSPFv3 协议其它参数
 - 1) 配置 OSPFv3 路由协议优先级
 - 2) 配置 OSPFv3 STUB 域及缺省路由的代价
 - 3) 配置 OSPFv3 虚链路
 - 4) 配置接口在选举指定三层交换机 DR 中的优先级

3. 关闭 OSPFv3 协议

1. 启动 OSPFv3 协议

在三层交换机上运行 OSPFv3 路由协议的基本配置也很简单，通常只需打开 OSPFv3 开关、在接口使能 OSPFv3 即可，OSPFv3 协议的参数都使用缺省值。如需修改 OSPFv3 协议参数值，参看 2.配置 OSPFv3 辅助参数。

命令	解释
全局配置模式	
[no] router IPv6 ospf <tag>	本命令初始化 ospfv3 路由进程并且进入 ospfv3 模式配置 ospfv3 路由进程。no 操作终止相应的进程。（必须）
OSPFv3 协议配置模式	
router-id <router_id> no router-id	为 ospfv3 进程配置路由器 ID。no 操作恢复 ID 到 0.0.0.0（必需）。
[no] passive-interface<ifname>	将一个接口设置为只收不发；本命令的 no 操作取消配置。
接口配置模式	

<pre>[no] IPv6 router ospf {area <area-id> [instance-id <instance-id> tag <tag> [instance-id <instance-id>]] tag <tag> area <area-id> [instance-id <instance-id>]}</pre>	在接口上使能 ospfv3 路由；本命令的 no 操作取消此配置。
--	-----------------------------------

2. 配置 OSPFv3 辅助参数

(1) 配置 OSPFv3 发包机制参数

1) 配置 OSPF 接口为只收不发

2) 配置接口发送数据包的代价

命令	解释
接口配置模式	
<pre>IPv6 ospf cost <cost> [instance-id <id>] no IPv6 ospf cost [instance-id <id>]</pre>	指定接口运行 OSPFv3 协议所需的代价；本命令的 no 操作恢复代价的缺省值。

3) 配置 OSPFv3 发包定时器参数（广播接口轮询发送 HELLO 数据包的定时器、邻接三层交换机失效定时器、接口传送 LSA 时延定时器、邻接三层交换机重传 LSA 定时器）

命令	解释
接口配置模式	
<pre>IPv6 ospf hello-interval <time> [instance-id <id>] no IPv6 ospf hello-interval [instance-id <id>]</pre>	配置接口周期发送 HELLO 数据包的时间间隔；本命令的 no 操作恢复为缺省值。
<pre>IPv6 ospf dead-interval <time> [instance-id <id>] no IPv6 ospf dead-interval [instance-id <id>]</pre>	配置认定相邻三层交换机失效的时间间隔；本命令的 no 操作恢复为缺省值。
<pre>IPv6 ospf transit-delay <time> [instance-id <id>] no IPv6 ospf transit-delay [instance-id <id>]</pre>	设置在接口上传送链路状态广播的时延值；本命令的 no 操作恢复为缺省值。
<pre>IPv6 ospf retransmit <time> [instance-id <id>] no IPv6 ospf retransmit [instance-id <id>]</pre>	配置接口与邻接三层交换机之间传送链路状态通告时的重传间隔；本命令的 no 操作恢复为缺省值。

(2) 配置 OSPFv3 引入路由参数

配置在 OSPFv3 中引入其它协议的路由

命令	解释
OSPF 协议配置模式	
[no]redistribute {kernel connected static rip isis bgp} [metric<value>] [metric-type {1 2}][route-map<word>]	引入其他协议发现路由和静态路由作为外部路由信息；本命令的 no 操作取消引入的外部路由信息。

(3) 配置 OSPFv3 引入其他 OSPFv3 进程路由功能

1) 启动 OSPFv3 引入其他 OSPFv3 进程路由功能

命令	解释
Router ipv6 ospf 配置模式	
redistribute ospf [<process-id>] [metric<value>] [metric-type {1 2}][route-map<word>] no redistribute ospf [<process-id>] [metric<value>] [metric-type {1 2}] [route-map<word>]	启动或者关闭 OSPFv3 引入其他 OSPFv3 进程路由功能。

2) 显示相关配置信息

命令	解释
特权模式或者配置模式	
show ipv6 ospf [<process-id>] redistribute	显示 OSPFv3 进程引入其它外部路由的配置信息。

3) 调试

命令	解释
特权模式	
debug ipv6 ospf redistribute message send no debug ipv6 ospf redistribute message send debug ipv6 ospf redistribute route receive no debug ipv6 ospf redistribute route receive	打开或关闭 OSPFv3 进程引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 OSPFv3 进程收到 nsm 发来的路由信息的调试开关。

(4) 配置 OSPFv3 协议其它参数

1) 配置 OSPF v3 STUB 域及缺省路由的代价

2) 配置 OSPFv3 虚链路

命令	解释
OSPFV3 协议配置模式	
timers spf <spf-delay> <spf-holdtime> no timers spf	配置 OSPFv3 的 SPF 定时器；本命令的 no 操作恢复为缺省值。
area <id> stub [no-summary] no area <id> stub [no-summary] area <id> default-cost <cost> no area <id> default-cost area <id> virtual-link A.B.C.D [instance-id <instance-id> INTERVAL] no area <id> virtual-link A.B.C.D [INTERVAL]	配置 OSPFv3 区域下的参数(STUB 域, 和虚链路)；本命令的 no 操作恢复为缺省值。

4) 配置接口在选举指定三层交换机 DR 中的优先级

命令	解释
接口配置模式	
IPv6 ospf priority <priority> [instance-id <id>] no IPv6 ospf priority [instance-id <id>]	配置接口在选举“指定三层交换机”时的优先级；本命令的 no 操作为恢复优先级缺省值。

3. 关闭 OSPFv3 协议

命令	解释
全局配置模式	
no router IPv6 ospf [<tag>]	关闭 OSPFv3 路由协议。

6.3 OSPFv3 案例

案例一：OSPFv3自治系统

以五台三层交换机为例组成一个OSPFv3自治系统，OSPF区域划分见下图：

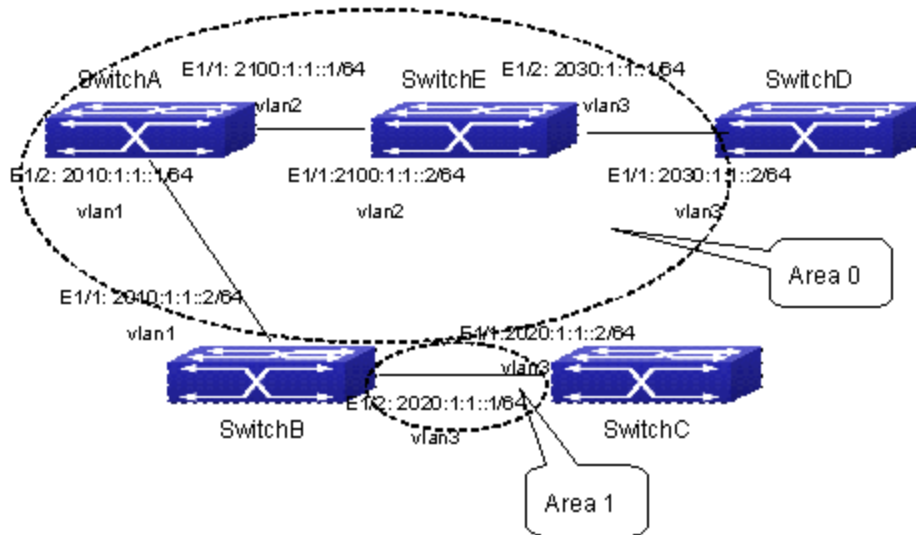


图 6-1 OSPFv3自治系统网络拓扑示意图

三层交换机Switch1- Switch5的配置分别如下：

三层交换机Switch1:

启动OSPFv3协议，配置路由器ID

```
Switch1(config)#router IPv6 ospf
```

```
Switch1 (config-router)#router-id 192.168.2.1
```

配置接口vlan1的IPv6地址和所属的OSPFV3域

```
Switch1#config
```

```
Switch1(config)# interface vlan 1
```

```
Switch1(config-if-vlan1)# IPv6 address 2010:1:1::1/64
```

```
Switch1(config-if-vlan1)# IPv6 router ospf area 0
```

```
Switch1(config-if-vlan1)#exit
```

配置接口vlan2的IP地址和所属的OSPFV3域

```
Switch1(config)# interface vlan 2
```

```
Switch1(config-if-vlan2)# IPv6 address 2100:1:1::1/64
```

```
Switch1(config-if-vlan2)# IPv6 router ospf area 0
```

```
Switch1 (config-if-vlan2)#exit
```

```
Switch1(config)#exit
```

```
Switch1#
```

三层交换机Switch2:

启动OSPFv3协议，配置路由器ID

```
Switch2(config)#router IPv6 ospf
```

```
Switch2 (config-router)#router-id 192.168.2.2
```

配置接口vlan1和vlan2的IPv6地址及所属的OSPFV3域。

```
Switch2#config
Switch2(config)# interface vlan 1
Switch2(config-if-vlan1)# IPv6 address 2010:1:1::2/64
Switch2(config-if-vlan1)# IPv6 router ospf area 0
Switch2(config-if-vlan1)#exit
Switch2(config)# interface vlan 3
Switch2(config-if-vlan3)# IPv6 address 2020:1:1::1/64
Switch2(config-if-vlan3)# IPv6 router ospf area 1
Switch2(config-if-vlan3)#exit
Switch2(config)#exit
Switch2#
```

三层交换机Switch3:

启动OSPFv3协议，配置路由器ID

```
Switch3(config)#router IPv6 ospf
Switch3(config-router)#router-id 192.168.2.3
```

配置接口vlan3的IPv6地址及所属的OSPFV3域。

```
Switch3#config
Switch3(config)# interface vlan 3
Switch3(config-if-vlan3)# IPv6 address 2020:1:1::2/64
Switch3(config-if-vlan3)# IPv6 router ospf area 1
Switch3(config-if-vlan3)#exit
Switch3(config)#exit
Switch3#
```

三层交换机Switch4:

启动OSPFv3协议，配置路由器ID

```
Switch4(config)#router IPv6 ospf
Switch4(config-router)#router-id 192.168.2.4
```

配置接口vlan3的IPv6地址及所属的OSPFV3域。

```
Switch4#config
Switch4(config)# interface vlan 3
Switch4(config-if-vlan3)# IPv6 address 2030:1:1::2/64
Switch4(config-if-vlan3)# IPv6 router ospf area 0
Switch4(config-if-vlan3)#exit
Switch4(config)#exit
Switch4#
```

三层交换机Switch5:

启动OSPFv3协议，配置路由器ID

```
Switch5(config)#router IPv6 ospf
```

```
Switch5(config-router)#router-id 192.168.2.5
配置接口vlan2的IPv6地址及所属的OSPFV3域。
Switch5#config
Switch5(config)# interface vlan 2
Switch5(config-if-vlan2)# IPv6 address 2100:1:1::2/64
Switch5(config-if-vlan2)# IPv6 router ospf area 0
Switch5(config-if-vlan2)#exit
配置接口vlan3的IPv6地址及所属的OSPFV3域
Switch5(config)# interface vlan 3
Switch5(config-if-vlan3)# IPv6 address 2030:1:1::1/64
Switch5(config-if-vlan3)# IPv6 router ospf area 0
Switch5(config-if-vlan3)#exit
Switch5(config)#exit
Switch5#
```

6.4 OSPFv3 排错帮助

在配置、使用 OSPFv3 协议时，可能会由于物理连接、配置错误等原因导致 OSPFv3 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误；
- ☞ 其次，保证接口和链路协议是 UP（使用 `show interface` 命令）；
- ☞ 然后在各接口上配置不同网段的 IPv6 地址；
- ☞ 然后，先启动 OSPFv3 协议（使用 `router IPv6 ospf` 命令）再在相应接口配置所属 OSPFv3 域；
- ☞ 接着，注意 OSPFv3 协议的自身特点——OSPFv3 骨干域（0 域）必须保证是连续的，如果不连续使用虚连接（`virtual link`）接来保证，所有非 0 域只能通过 0 域与其他非 0 域相连，不允许非 0 域直接相连；边界三层交换机是指该三层交换机的一部分接口属于 0 域，而另外一部分接口属于非 0 域；对于广播网等多访问网，需要选举指定三层交换机 DR；对于每个 `ospfv3` 进程必须配置不是 0.0.0.0 的路由器 ID。

第7章 BGP

7.1 BGP介绍

BGP（Border Gateway Protocol）是边界网关协议。它是一种自治系统（AS，autonomous system）间的动态路由发现协议。它的基本功能是在AS间自动交换无环路的路由信息。BGP通过交换带有自治系统编号的AS序列属性的路径可达信息，来选择最佳路由从而消除路由环路并实施用户配置的策略。通常在一个AS内部的交换机可以使用多种内部网关协议（IGP，Interior Gateway Protocol）来相互交换AS内部的路由信息，如RIP、OSPF都是IGP协议；而用外部网关协议（EGP，Exterior Gateway Protocol）来交换AS之间的路由信息，如BGP就是EGP中的一种。通常一个AS是建立在单一的管理部门之上的。BGP经常用于ISP之间或是跨国公司的各个分部之间的交换机上。

BGP从1989年就开始使用，它最早发布的三个版本分别是RFC1105（BGP-1）、RFC1163（BGP-2）和RFC1267（BGP-3），当前最流行的是使用的RFC1771（BGP-4）。本交换机目前支持BGP-4。

1. BGP-4的特性

BGP-4适用于分布式结构并支持无类域间路由（CIDR，Classless InterDomain Routing）。BGP-4正迅速成为事实上的Internet外部路由协议标准，BGP-4特性如下：

- BGP是一种外部路由协议，与OSPF、RIP等的内部路由协议不同，BGP并不能发现和计算路由，但可以控制路由的传播和选择最好的路由。
- 通过在更新路由中携带AS路径信息可以彻底解决路由环路的问题。
- 使用TCP作为其传输层协议提高了协议的可靠性。端口号为179。
- BGP-4支持无类域间路由（CIDR，Classless InterDomain Routing），这是较BGP-3的一个重要改进。CIDR以一种全新的方法看待IP地址，不再区分A类网、B类网及C类网。例如一个非法的C类网络地址192.213.0.0 255.255.0.0 采用CIDR表示法192.213.0.0/16就成为一个合法的超级网络。其中/16表示网络号由从地址左端开始的16比特构成。CIDR的引入简化了路由聚合（Routes Aggregation）。路由聚合实际上是合并几个不同路由的过程，这样从通告几条路由变为通告一条路由，减化了路由表。
- 路由更新时BGP只发送增量路由，大大减少了BGP传播路由所占用的带宽，适用于在Internet上传播大量的路由信息。
- 由于政治上和经济上的原因，每个自治系统希望对路由进行过滤选择和控制，因此BGP-4提供了丰富的路由策略，它使得BGP便于扩展以支持因特网新的发展。

2. BGP-4运行概述

不同于RIP和OSPF协议，BGP协议是面向连接的。BGP交换机之间要交换路由信息就一定需要建立连接后才能交换路由信息。BGP协议的运行是通过消息驱动的，其消息共可分为四类：

打开消息(Open Message)——是TCP连接建立后第一个发送的消息。它用于建立BGP同伴间的BGP连接关系。Open消息中的一些参数用于协商BGP邻居间的连接能否建立。

保持消息(Keepalive Message)——是用于检测连接有效性的消息。一般是周期发送用于保持BGP连接。若在holdtime时间内没有收到这个消息或者Update消息，则关闭BGP连接。

更新消息(Update Message)——是BGP系统中最重要的消息，用于在同伴之间交换路由信息。除了交换机交换更新的路由信息，也交换不可用的或是已经取消的路由信息。它由三部分构成：不可达路由(unreachable)、网络可达性信息(NLRI Network Layer Reachability Information)、路径属性(Path Attributes)。

通告消息(Notification Message)——是错误通告消息。当一个BGP发言人收到这样的消息时要关闭与它的邻居的BGP连接。

BGP-4是面向连接的。BGP系统作为高层协议，运行在一个特定的网络设备上。当发现邻居后，就建立并维护一个TCP会话。建立了会话之后就要进行路由表的交换和同步。仅仅在系统初启时，通过发送整个BGP路由表交换路由信息。之后只有在有更新的路由信息时，才交换路由信息。此时只交换增量更新信息(update message)。BGP-4通过使用keepalive信息来周期性地维护链路和会话，即通过定期地接收和发送保活消息(keepalive)来检测相互之间的连接是否正常。

参与BGP会话的交换机称为BGP发言者(speaker)。它不断的接收或产生新路由信息，并将它通告(advertise)给其它的BGP发言者。当BGP发言者收到来自其它自治系统的新路由通告时，如果该路由比当前已知路由好，或者当前还没有可接受路由，那么它就把这个路由通告给自治系统内所有其它的BGP发言者。一个BGP发言者也将同它交换消息的其它的BGP发言者称为邻居(neighbor)或对等体(peer)，若干相关的邻居可以构成对等体组(peer group)。BGP在交换机上以下列两种方式运行：

- 1、IBGP: Internal BGP
- 2、EBGP: External BGP

当BGP运行于同一AS内部时被称为IBGP。当BGP运行于不同自治系统之间时称为EBGP。通常，外部邻居彼此物理上相邻，而内部邻居则可以在同一AS的任何地方。它们之间的不同最终体现在BGP路由协议对路由信息的处理方式上。设备可以通过检查邻居交换机发送的打开消息中的AS号，来决定邻居的BGP交换机是作为外部邻居还是作为内部邻居。

IBGP在AS内部使用，它向该AS内的所有BGP邻居传递信息。IBGP在一个大的组织内交流AS路由信息。注意，AS内的交换机不必是物理介质上相邻，只要是交换机位于同一个AS内部，那么两者之间就可以互为邻居。因为BGP本身不能发现路由，因此需要其它的内部路由协议(如静态路由、直连路由、OSPF和RIP)的路由表中包含邻居的IP地址所在的网段。并用该路由进行BGP之间的信息交换。为了防止路由环路，一个BGP发言人从内部邻居收到路由通告时，不会将这些路由通告给其它内部邻居。

EBGP在AS之间使用，它向外部AS的BGP邻居传递路由信息。EBGP之间一般彼此物理上临近，即共享同一介质。因为EBGP之间要物理上相互临近，所以在两个AS之间的边界设备通常就是EBGP。一个BGP发言人从外部邻居收到路由通告时，则将这些路由通告给其它内部邻居。

3. 路由属性

BGP-4可以通过相关机制达到共享和查询内部IP路由表的目的，但它仍然拥有自己的路由表。在BGP路由表中，每一条路由都拥有：一个网络号（network number），所经过的AS列表信息（也被称为as路径），和一些路由属性（如origin）。BGP-4使用的路由属性很复杂，该属性可以在进行路径选择时提供度量的依据。

4. BGP的路由选择策略

当从多个邻居收到关于同一个路由的BGP通告时，经过路由过滤后需要考虑选择最优路由。这个过程叫做BGP的路由选择过程。BGP的路由选择过程只有下列条件满足时才会开始：

1. 交换机的路由必需是下一跳可达的。即在路由表中，有能够到达下一跳地址的路由。
2. BGP 必需和 IGP 同步（除非设置成了非同步，仅限于 IBGP）

BGP路由选择过程基于BGP的属性值。当有多条路由是指明同一个目的地时，BGP则要选择一条到达目的地的最佳路由。该决策过程如下：

1. 优先选用具有最大权重（weight）的路由；
2. 如果权重相同，优选具有最大本地优先级（local preference）的路由；
3. 如果本地优先级相同，优选本地交换机始发的路由。
4. 如果本地优先级相同，并且没有本地交换机始发的路由，优选具有最短AS路径的路由；
5. 如果 AS 路径相同，优选具有最低 origin 类型的路由（IGP<EGP<INCOMPLETE）；
6. 如果 origin 类型相同，优选具有最低 MED 属性的路由，除非激活 bgp always-compare-med命令，否则这种比较只在来自同一邻居AS的路由间进行。
7. 如果 MED 属性相同，EBGP 优于联盟外部，联盟外部优于 IBGP；
8. 如果所有前面的条件都相同，则判断这些路由的 NEXT_HOP 属性，在 IGP 路由表中到达其下一跳地址对应的路由 cost 最低的路由被优选
9. 如果至此仍然相同，则用BGP交换机ID（router ID）打破平衡。最优路由来自于交换机的BGP Router-ID 最小的那个。

7.2 BGP配置

BGP 的配置任务分为基本配置任务和高级配置任务。BGP 的基本的配置任务包括：

- 1. 启动 BGP 路由（必须）
- 2. 配置 BGP 邻居（必须）
- 3. 管理路由策略的改变
- 4. 配置 BGP 权重
- 5. 设置基于邻居的 BGP 路由过滤策略
- 6. 设置 BGP 的下一跳
- 7. 配置 EBGP 的多跳
- 8. 配置 BGP 的会话标识
- 9. 配置 BGP 的版本号

高级 BGP 配置任务包括：

- 1. 使用路由映射（route-map）来更改路由
- 2. 配置路由聚合
- 3. 配置路由的团体属性过滤
- 4. 设置 BGP 的联盟
- 5. 设置路由反射器
- 6. 设置对等体组
- 7. 配置邻居和对等体组参数
- 8. 调整 BGP 定时器
- 9. 调整 BGP 的通告间隔
- 10. 设置默认的本地优先级
- 11. 允许发送缺省路由
- 12. 设置 BGP 的 MED 值
- 13. 配置 BGP 的路由重分配
- 14. 配置 BGP 路由衰减
- 15. 配置 BGP 能力协商
- 16. 配置路由服务器
- 17. 配置选路规则
- 18. 配置 BGP 引入不同进程 OSPF 路由
 - （1）启动 BGP 引入不同进程 OSPF 路由功能
 - （2）显示和调试 BGP 引入不同进程 OSPF 路由功能相关信息

一. 基本 BGP 配置任务

1. 启动 BGP 路由

命令	解释
全局配置模式	

router bgp <as-id> no router bgp <as-id>	启动 BGP，该命令的 no 形式关闭 BGP 进程。
BGP 协议配置模式	
bgp asnotation asdot no bgp asnotation asdot	配置以分隔符（asdot）方式显示 AS 号 and 进行正则表达式匹配。
network <ip-address/M> no network <ip-address/M>	设置 BGP 要通告的网络，该命令的 no 形式取消要通告的网络。
address-family ipv4 {unicast multicast vrf <vrf-nam>} no address-family ipv4 {unicast multicast vrf <vrf-nam>}	创建 BGP 协议 IPv4 并进入 BGP-VPN 实例视图。缺省情况下未创建任何 IPv4。

2. 配置 BGP 邻居

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} remote-as <as-id> no neighbor {<ip-address> <TAG>} [remote-as <as-id>]	指定一个 BGP 邻居，该命令的 no 操作删除邻居。

3. 管理路由策略的改变

(1) 配置硬重建

命令	解释
特权用户模式	
clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>}	配置 BGP 硬重建。

(2) 配置出站软重建

命令	解释
特权用户模式	
clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>} soft out	配置 BGP 出站软重建。

(3) 配置入站软重建

命令	解释
BGP 协议配置模式	

neighbor { <ip-address> <TAG> } soft-reconfiguration inbound no neighbor { <ip-address> <TAG> } soft-reconfiguration inbound	该命令可以存储从邻居或是对等体组发过来的路由信息，该命令的 no 形式取消路由信息的存储。
特权用户模式	
clear ip bgp {<*> <as-id> external peer-group <NAME> <ip-address>} soft in	配置 BGP 入站软重建。

4. 配置 BGP 权重

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} weight <weight> no neighbor {<ip-address> <TAG>} weight <weight>	配置 BGP 邻居的权重值，该命令的 no 形式恢复默认的权重值。

5. 设置基于邻居的 BGP 路由过滤策略

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} distribute-list {<1-199> <1300-2699> <WORD>} {in out} no neighbor {<ip-address> <TAG>} distribute-list {<1-199> <1300-2699> <WORD>} {in out}	过滤邻居的路由更新信息，该命令的 no 形式取消路由过滤。

6. 设置 BGP 的下一跳

1) 设置下一跳为本交换机的地址

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} next-hop-self no neighbor {<ip-address> <TAG>} next-hop-self	发送路由时下一跳设置为自己的地址。该命令的 no 形式取消设置。

2) 通过路由映射取消默认的下一跳

命令	解释
路由映射配置模式	
set ip next-hop <ip-address> no set ip next-hop	设置出去路由的下一跳属性，该命令的 no 形式取消该设置。

7. 配置 EBGp 的多跳

如果外部邻居之间的连接不是直连的，那么可以使用下面的命令配置邻居的多跳。

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>] no neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>]	配置允许同不直接相连网络上的交换机建立 EBGp 连接。该命令的 no 形式取消设置。

8. 配置 BGP 的会话标识

命令	解释
BGP 协议配置模式	
bgp router-id <ip-address> no bgp router-id	配置 router-id 标识的值，该命令的 no 形将恢复默认值。

9. 配置 BGP 的版本号

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} version <value> no neighbor {<ip-address> <TAG>} version	设置 BGP 邻居使用的版本号，该命令的 no 形式恢复缺省设置。目前仅支持版本 4。

二. 高级 BGP 配置任务

1. 使用路由映射（route-map）来更改路由

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} route-map <map-name> {in out} no neighbor {<ip-address> <TAG>} route-map <map-name> {in out}	将路由映射应用于入站或出站的路由。no 命令取消路由映射的设置。

2. 配置路由聚合

命令	解释
BGP 协议配置模式	

aggregate-address <i><ip-address/M></i> [summary-only] [as-set] no aggregate-address <i><ip-address/M></i> [summary-only] [as-set]	在 BGP 路由表中建立一条聚合记录，该命令的 no 形式取消聚合记录。
---	--------------------------------------

3. 配置路由的团体属性过滤

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} send-community no neighbor {<ip-address> <TAG>} send-community	允许向 BGP 邻居发送的路由更新带有团体属性，该命令的 no 形式使向邻居发送的路由不带团体属性。

4. 设置 BGP 的联盟

命令	解释
BGP 协议配置模式	
bgp confederation identifier <as-id> no bgp confederation identifier <as-id>	配置一个 BGP 自治系统联盟标识，该命令的 no 形式删除 BGP 自治系统联盟标识。
bgp confederation peers <as-id> [<as-id>..] no bgp confederation peers <as-id> [<as-id>..]	配置属于自治系统联盟的 AS，该命令的 no 形式从自治系统联盟中删除 AS。

5. 设置路由反射器

(1) 设置路由反射器和它的客户端可以使用下面的命令：

命令	解释
BGP 协议配置模式	
neighbor <ip-address> route-reflector-client no neighbor <ip-address> route-reflector-client	配置当前的交换机做为路由反射器并且指明一个客户端，该命令的 no 形式删除一个客户端。

(2) 如果簇内有多于一个路由反射器，设置簇 ID 的命令如下所示：

命令	解释
BGP 协议配置模式	
bgp cluster-id <cluster-id> no bgp cluster-id	配置簇 ID，该命令的 no 形式簇 ID 的设置。

(3) 如果需要进行客户端到客户端的路由反射，那么可以用下面的命令：

命令	解释
BGP 协议配置模式	
bgp client-to-client reflection no bgp client-to-client reflection	配置允许客户端到客户端的路由反射, 该命令的 no 形式禁止客户端到客户端的路由反射。

6. 设置对等体组

(1) 创建对等体组

命令	解释
BGP 协议配置模式	
neighbor <TAG> peer-group no neighbor <TAG> peer-group	创建对等体组, 该命令的 no 形式删除对等体组。

(2) 将邻居加入对等体组内

命令	解释
BGP 协议配置模式	
neighbor <ip-address> peer-group <TAG> no neighbor <ip-address> peer-group <TAG>	配置向对等体组内添加一个成员, 该命令的 no 形式取消指定的成员。

7. 配置邻居和对等体组参数

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} remote-as <as-id> no neighbor {<ip-address> <TAG>} remote-as <as-id>	指定一个 BGP 邻居, 该命令的 no 操作删除邻居。
neighbor {<ip-address> <TAG>} description <.LINE> no neighbor {<ip-address> <TAG>} description	指定和 BGP 邻居的描述信息, 该命令的 no 形式删除描述信息。
neighbor {<ip-address> <TAG>} default-originate [route-map <NAME>] no neighbor {<ip-address> <TAG>} default-originate [route-map <NAME>]	允许发送缺省路由 0.0.0.0, 该命令的 no 形式取消发送默认路由。
neighbor {<ip-address> <TAG>} send-community	设置向邻居发送路由的团体属性。

no neighbor {<ip-address> <TAG>} send-community	
neighbor {<ip-address> <TAG>} timers <keepalive> <holdtime> no neighbor {<ip-address> <TAG>} timers	设置某个特定的邻居的 keepalive 和 holdtime 定时器时间, 该命令的 no 形式恢复默认值。
neighbor {<ip-address> <TAG>} advertisement-interval <seconds> no neighbor {<ip-address> <TAG>} advertisement-interval	设置发送 BGP 路由信息之间的最小间隔, 该命令的 no 形式恢复默认值。
neighbor {<ip-address> <TAG>} ebgp-multihop [<1-255>] no neighbor {<ip-address> <TAG>} ebgp-multihop	配置允许同不直接相连网络上的 EBGP 建立连接。该命令的 no 形式取消设置。
neighbor {<ip-address> <TAG>} weight <weight> no neighbor {<ip-address> <TAG>} weight	配置 BGP 邻居的权重值, 该命令的 no 形式恢复默认的权重值。
neighbor {<ip-address> <TAG>} distribute-list { <access-list-number> <name> } { in out } no neighbor {<ip-address> <TAG>} distribute-list { <access-list-number> <name> } { in out }	过滤邻居的路由更新信息, 该命令的 no 形式取消路由过滤。
neighbor {<ip-address> <TAG>} route-reflector-client no neighbor {<ip-address> <TAG>} route-reflector-client	配置当前的交换机做为路由反射器并且指明一个客户端, 该命令的 no 形式删除一个客户端。
neighbor {<ip-address> <TAG>} next-hop-self no neighbor {<ip-address> <TAG>} next-hop-self	发送路由时下一跳设置为自己的地址。该命令的 no 形式取消设置。
neighbor {<ip-address> <TAG>} version <value> no neighbor {<ip-address> <TAG>} version	指定和 BGP 邻居进行通信的 BGP 版本, 该命令的 no 形式恢复默认设置。
neighbor {<ip-address> <TAG>} route-map <map-name> { in out } no neighbor {<ip-address> <TAG>} route-map <map-name> { in out }	将路由映射应用于入站或出站的路由。no 命令取消路由映射的设置。
neighbor {<ip-address> <TAG>}	该命令可以存储从邻居或是对等体

soft-reconfiguration inbound no neighbor {<ip-address> <TAG>} soft-reconfiguration inbound	组发过来的路由信息，该命令的 no 形式取消路由信息的存储。
neighbor {<ip-address> <TAG>} shutdown no neighbor {<ip-address> <TAG>} shutdown	关闭 BGP 邻居或对等体组，该命令的 no 形式激活已经关闭的 BGP 邻居或对等体组。

8. 调整 BGP 定时器

(1) 设置所有邻居的 BGP 定时器。

命令	解释
BGP 协议配置模式	
timers bgp <keepalive> <holdtime> no timers bgp	设置所有邻居的 BGP 定时器，该命令的 no 形式恢复默认值。

(2) 设置特定邻居的定时器值

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} timers <keepalive> <holdtime> no neighbor {<ip-address> <TAG>} timers	设置某个特定的邻居的 keepalive 和 holdtime 定时器时间。该命令的 no 形式恢复默认值。

9. 调整 BGP 的通告间隔

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} advertisement-interval <seconds> no neighbor {<ip-address> <TAG>} advertisement-interval	设置发送 BGP 路由更新信息之间的最小间隔，该命令的 no 形式恢复缺省设置。

10. 设置默认的本地优先级

命令	解释
BGP 协议配置模式	
bgp default local-preference <value> no bgp default local-preference	改变默认的本地优先级，该命令的 no 形式恢复默认值。

11. 允许发送缺省路由

命令	解释
----	----

BGP 协议配置模式	
neighbor {<ip-address> <TAG>} default-originate no neighbor {<ip-address> <TAG>} default-originate	允许发送缺省路由 0.0.0.0，该命令的 no 形式取消发送默认路由

12. 设置 BGP 的 MED 值

(1) 设置 MED 值

命令	解释
路由映射配置模式	
set metric <metric-value> no set metric	配置 metric 的值，该命令的 no 形式恢复默认值

(2) 对来自不同 AS 的路径应用基于 MED 的路径选择

命令	解释
BGP 协议配置模式	
bgp always-compare-med no bgp always-compare-med	允许对来自不同 AS 的路径进行 MED 值的比较，该命令的 no 形式禁止对来自不同 AS 的路径进行 MED 值的比较。

13. 配置 BGP 的路由重分配

命令	解释
BGP 协议配置模式	
redistribute {connected static rip ospf} [metric <metric>] [route-map <NAME>] no redistribute {connected static rip ospf}	将 IGP 路由重分配到 BGP。同时可以指定再分配的 metric、路由映射，该命令的 no 形式取消重分配。

14. 配置 BGP 的路由衰减

命令	解释
BGP 协议配置模式	
bgp dampening [<1-45>] [<1-20000> <1-20000> <1-255>] [<1-45>] no bgp dampening	启动 BGP 路由衰减并使用设定的参数，其 no 形式停止路由衰减。

15. 配置 BGP 的能力协商

命令	解释
----	----

BGP 协议配置模式	
neighbor {<ip-address> <TAG>} capability {dynamic route-refresh} no neighbor {<ip-address> <TAG>} capability {dynamic route-refresh} neighbor {<ip-address> <TAG>} capability orf prefix-list {<both> <send> <receive>} no neighbor {<ip-address> <TAG>} capability orf prefix-list {<both> <send> <receive>} neighbor {<ip-address> <TAG>} dont-capability-negotiate no neighbor {<ip-address> <TAG>} dont-capability-negotiate neighbor {<ip-address> <TAG>} override-capability no neighbor {<ip-address> <TAG>} override-capability neighbor {<ip-address> <TAG>} strict-capability-match no neighbor {<ip-address> <TAG>} strict-capability-match	BGP 提供能力协商机制，在建立连接时进行能力的匹配。目前支持的能力包括路由刷新、动态能力、出路由过滤（ORF）能力以及地址族支持协商的能力。使用这些命令来使能这些能力，其 NO 形式关闭这些能力。也可以通过命令来配置不进行能力协商，进行严格的能力协商或不管协商的结果。

16. 配置路由服务器

命令	解释
BGP 协议配置模式	
neighbor {<ip-address> <TAG>} route-server-client no neighbor {<ip-address> <TAG>} route-server-client	路由服务器可以在 EBGp 环境下集中配置 BGP 邻居，从而减少每个客户配置的对等体的个数，本命令配置本机为路由服务器并指定其服务的客户，其 NO 形式可删除客户。

17. 配置选路规则

命令	解释
BGP 协议配置模式	

bgp always-compare-med no bgp always-compare-med bgp bestpath as-path ignore no bgp bestpath as-path ignore bgp bestpath compare-confed-asp no bgp bestpath compare-confed-asp bgp bestpath compare-routerid no bgp bestpath compare-routerid bgp bestpath med {[confed] [missing-is-worst]} no bgp bestpath med {[confed] [missing-is-worst]}	BGP 可以通过设置改变一些选路规则，从而达到改变最优选择的目的，可以通过这些命令使在 EBGP 环境下比较 MED，忽视 AS-PATH 的长度，比较联盟的 as-path 长度，比较路由器标识，比较联盟 MED 等规则，通过其 NO 形式恢复默认的选路规则。
---	---

18. 配置 BGP 引入不同进程 OSPF 路由

(1) 启动 BGP 引入不同进程 OSPF 路由功能

命令	解释
Router bgp 配置模式	
redistribute ospf [<process-id>] [route-map<word>] no redistribute ospf [<process-id>]	启动或者关闭 BGP 引入其他 OSPF 进程路由功能。

(2) 显示和调试 BGP 引入不同进程 OSPF 路由功能相关信息

命令	解释
特权和配置模式	
show ip bgp redistribute	显示 BGP 进程引入其它外部路由的配置信息。
特权模式	
debug bgp redistribute message send no debug bgp redistribute message send debug bgp redistribute route receive no debug bgp redistribute route receive	打开或关闭 BGP 引入其他 OSPF 进程路由的命令下发调试开关。 打开或关闭 BGP 进程收到 nsm 发来的路由信息的调试开关。

7.3 BGP典型案例

7.3.1 案例一：BGP邻居配置

SwitchB、SwitchC和SwitchD位于AS200中，SwitchA位于AS100中。SwitchA和SwitchB共享一个相同的网络段11.0.0.0。而SwitchB和SwitchD彼此物理上不相邻。

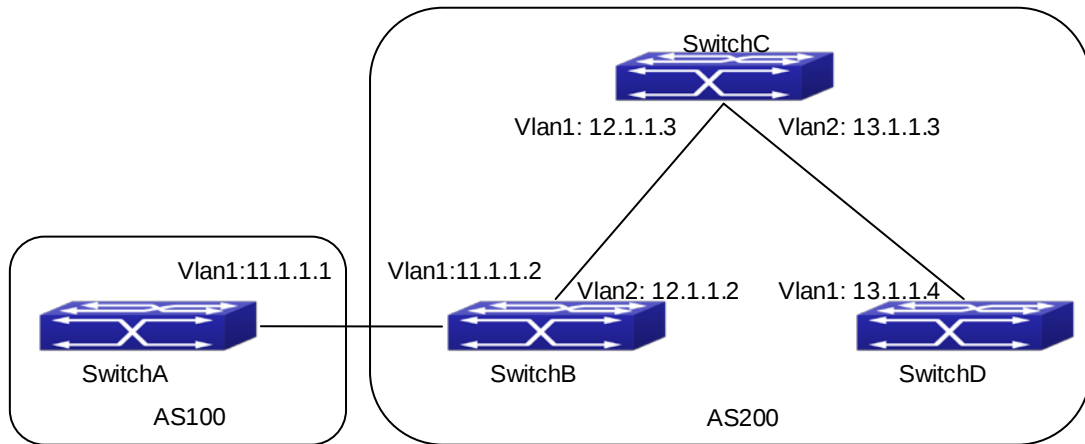


图 7-1 BGP 网络拓扑图

则SwitchA的配置如下：

```
SwitchA(config)#router bgp 100
SwitchA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
SwitchA(config-router-bgp)#exit
```

SwitchB的配置如下：

```
SwitchB(config)#router bgp 200
SwitchB(config-router-bgp)#network 11.0.0.0
SwitchB(config-router-bgp)#network 12.0.0.0
SwitchB(config-router-bgp)#network 13.0.0.0
SwitchB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
SwitchB(config-router-bgp)#neighbor 12.1.1.3 remote-as 200
SwitchB(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchB(config-router-bgp)#exit
```

SwitchC的配置如下：

```
SwitchC(config)#router bgp 200
SwitchC(config-router-bgp)#network 12.0.0.0
SwitchC(config-router-bgp)#network 13.0.0.0
SwitchC(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
SwitchC(config-router-bgp)#neighbor 13.1.1.4 remote-as 200
SwitchC(config-router-bgp)#exit
```

RouteD的配置如下：

```
SwitchD(config)#router bgp 200
SwitchD(config-router-bgp)#network 13.0.0.0
SwitchD(config-router-bgp)#neighbor 12.1.1.2 remote-as 200
SwitchD(config-router-bgp)#neighbor 13.1.1.3 remote-as 200
SwitchD(config-router-bgp)#exit
```

此时SwitchB和SwitchA之间的连接是EBGP，和SwitchC、SwitchD之间的连接是IBGP。

SwitchB和SwitchD之间不必物理上相连就可以进行BGP的连接，但前提是两个交换机必须有到达对方的路由。该路由可以通过静态路由或是IGP获得。

7.3.2 案例二：BGP聚合配置

在这个例子中，设置路由聚合。首先使用redistribute命令将静态路由重分配到BGP路由表中：

```
SwitchB(config)#ip route 193.0.0.0/24 11.1.1.1
SwitchB(config)#router bgp 100
SwitchB(config-router-bgp)#redistribute static
```

当路由表中至少有一条路由属于指定范围时，下面的配置就在BGP路由表中创建一个聚合路由。聚合路由将被认为来自本机产生的AS。而关于193.0.0.0的更具体的路由信息也进行通告：

```
SwitchB(config)#router bgp 100
SwitchB(config-router-bgp)#aggregate 193.0.0.0/16
此时也可以将上面的aggregate命令改为下面的命令，则此时该交换机只通告聚合路由
193.0.0.0，而禁止把更具体的路由通告给所有邻居：
SwitchB(config-router-bgp)#aggregate 193.0.0.0/16 summary-only
```

7.3.3 案例三：配置BGP团体属性

下面的例子中，route map set-community用于到邻居16.1.1.6的出站更新。此时通过访问列表1的路由设置特殊的团体属性值“1111”，其它的路由进行正常通告。

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map set-community out
Switch(config-router-bgp)#exit
Switch(config)#route-map set-community permit 10
Switch(config-route-map)#match ip address 1
Switch(config-route-map)#set community 1111
```

```
Switch(config-route-map)#exit
Switch(config)#route-map set-community permit 20
Switch(config-route-map)#match ip address 2
Switch(config-route-map)#exit
Switch(config)#access-list 1 permit 11.1.0.0 0.0.255.255
Switch(config)#access-list 2 permit 0.0.0.0 255.255.255.255
Switch(config)#exit
```

```
Switch#clear ip bgp 16.1.1.6 soft out
```

下面的例子中，根据路由的团体属性值有选择性地设置来自邻居16.1.1.6的路由的MED和本地优先级值。所有同团体列表com1匹配的路由都设置MED为2000，团体列表com1允许团体值带有“100 200 300”或“900 901”的路由通过。同时该路由也可能具有其它团体属性。所有通过团体列表com2的路由都设置本地优先级值为500。而如果既不能通过“com1”，也不能通过“com2”的团体列表的路由将被拒绝。

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#neighbor 16.1.1.6 remote-as 200
Switch(config-router-bgp)#neighbor 16.1.1.6 route-map match-community in
Switch(config-router-bgp)#exit
Switch(config)#route-map match-community permit 10
Switch(config-route-map)#match community com1
Switch(config-route-map)#set metric 2000
Switch(config-route-map)#exit
Switch(config)#route-map match-community permit 20
Switch(config-route-map)#match community com2
Switch(config-route-map)#set local-preference 500
Switch(config-route-map)#exit
Switch(config)#ip community-list com1 permit 100 200 300
Switch(config)#ip community-list com1 permit 900 901
Switch(config)#ip community-list com2 permit 88
Switch(config)#ip community-list com2 permit 90
Switch(config)#exit
Switch#clear ip bgp 16.1.1.6 soft out
```

7.3.4 案例四：BGP联盟配置

下面是一个自治系统联盟的配置。如图所示，SWB、SWC 建立 IBGP 连接，属于自治系统 10；SWD 属于自治系统 20；SWB 与 SWD 建立自治系统联盟内部 EBGP 连接；AS10、AS20 组成自治系统联盟，自治系统号为 AS200；SWA 属于自治系统 AS100，SWB 通过 SWA 与自治系统 200 建立 EBGP 连接。

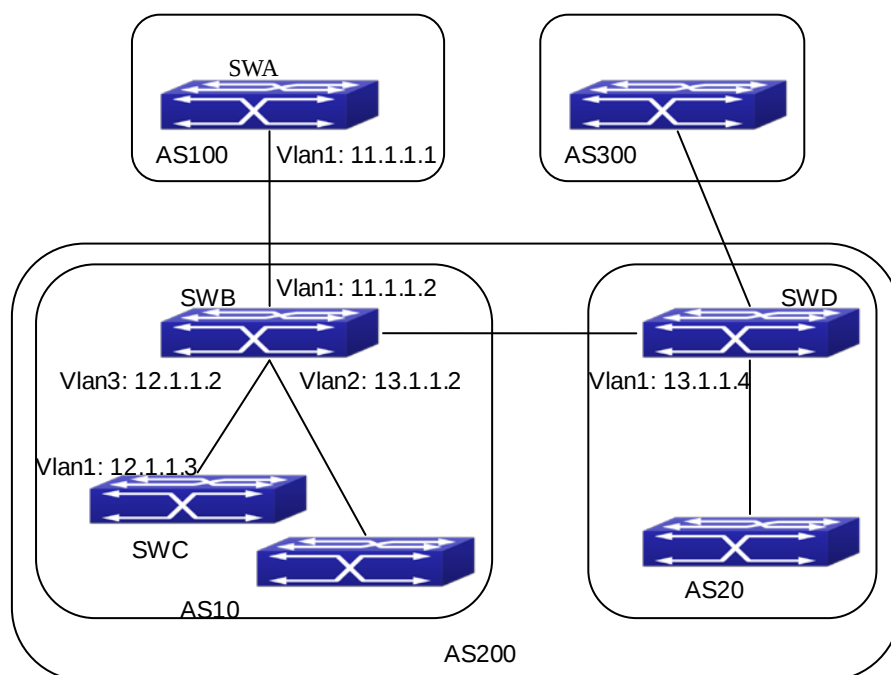


图 7-2 联盟配置拓扑图

配置如下：

SWA:

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

```
SWA(config-router-bgp)#neighbor 11.1.1.2 remote-as 200
```

SWB:

```
SWB(config)#router bgp 10
```

```
SWB(config-router-bgp)#bgp confederation identifier 200
```

```
SWB(config-router-bgp)#bgp confederation peers 20
```

```
SWB(config-router-bgp)#neighbor 12.1.1.3 remote-as 10
```

```
SWB(config-router-bgp)#neighbor 13.1.1.4 remote-as 20
```

```
SWB(config-router-bgp)#neighbor 11.1.1.1 remote-as 100
```

SWC:

```
SWC(config)#router bgp 10
```

```
SWC(config-router-bgp)#bgp confederation identifier 200
```

```
SWC(config-router-bgp)#bgp confederation peers 20
```

```
SWC(config-router-bgp)#neighbor 12.1.1.2 remote-as 10
```

SWD:

```
SWD(config)#router bgp 20
```

```
SWD(config-router-bgp)#bgp confederation identifier 200
```

```
SWD(config-router-bgp)#bgp confederation peers 10
```

```
SWD(config-router-bgp)#neighbor 13.1.1.2 remote-as 10
```

7.3.5 案例五：BGP路由反射器配置

下面是一个路由反射器的配置，如图所示，SWA、SWB、SWC、SWD、SWE、SWF 和 SWG 建立 IBGP 连接，属于 AS100。SWC 和 AS200 建立 EBGP 连接。SWA 和 AS300 建立 EBGP 连接。SWC、SWD 和 SWG 作路由反射器。

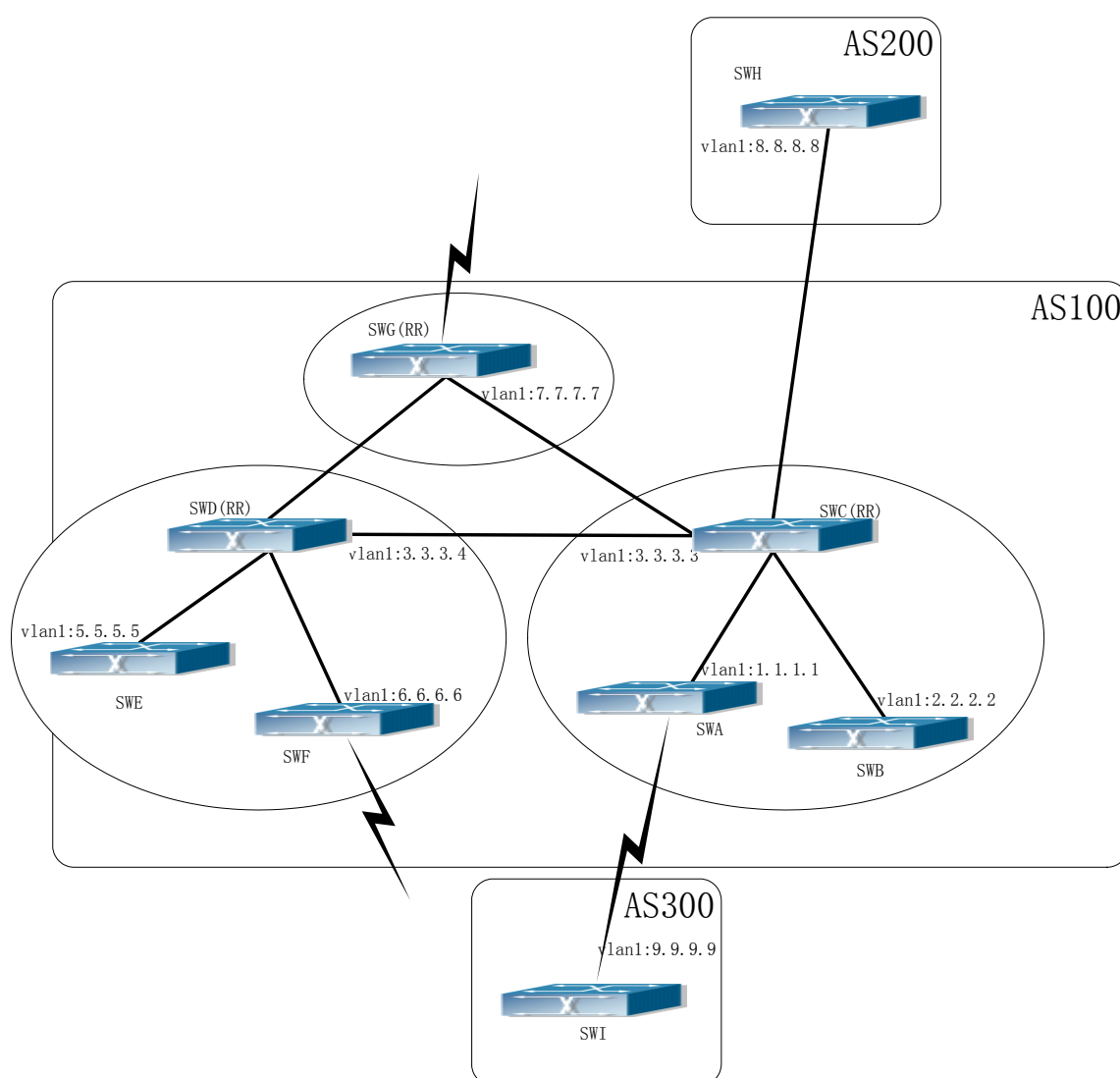


图 7-3 路由反射器拓扑图

配置如下：

SWC 的配置：

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

```
SWC(config-router-bgp)#neighbor 1.1.1.1 remote-as 100
SWC(config-router-bgp)#neighbor 1.1.1.1 route-reflector-client
SWC(config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SWC(config-router-bgp)#neighbor 2.2.2.2 route-reflector-client
SWC(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
SWC(config-router-bgp)#neighbor 3.3.3.4 remote-as 100
SWC(config-router-bgp)#neighbor 8.8.8.8 remote-as 200
```

SWD 的配置:

```
SWD(config)#router bgp 100
SWD(config-router-bgp)#neighbor 5.5.5.5 remote-as 100
SWD(config-router-bgp)#neighbor 5.5.5.5 route-reflector-client
SWD(config-router-bgp)#neighbor 6.6.6.6 remote-as 100
SWD(config-router-bgp)#neighbor 6.6.6.6 route-reflector-client
SWD(config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SWD(config-router-bgp)#neighbor 7.7.7.7 remote-as 100
```

SWA 的配置:

```
SWA(config)#router bgp 100
SWA(config-router-bgp)#neighbor 1.1.1.2 remote-as 100
SWA(config-router-bgp)#neighbor 9.9.9.9 remote-as 300
```

此时的 SWA 不需要和所有的 AS100 的交换机建立 IBGP 连接,就可以收到本 AS 内的其它交换机的 BGP 路由。

7.3.6 案例六: BGP的MED设置

下面是一个 MED 的配置,如图所示,SWA 属于 AS100,SWB 属于 AS400,SWC 和 SWD 属于 AS300。

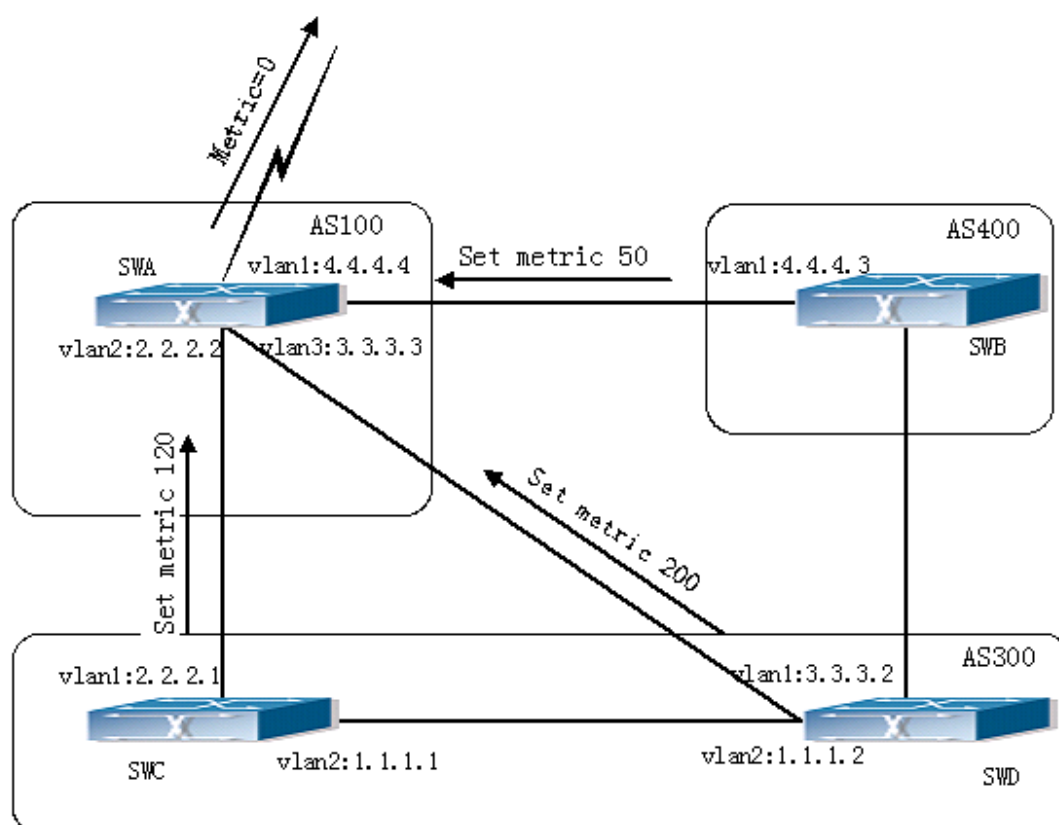


图 7-4 MED 配置拓扑图

SWA 的配置

```
SWA(config)#router bgp 100
SWA(config-router-bgp)#neighbor 2.2.2.1 remote-as 300
SWA(config-router-bgp)#neighbor 3.3.3.2 remote-as 300
SWA(config-router-bgp)#neighbor 4.4.4.3 remote-as 400
```

SWC 的配置

```
SWC(config)#router bgp 300
SWC (config-router-bgp)#neighbor 2.2.2.2 remote-as 100
SWC (config-router-bgp)#neighbor 2.2.2.2 route-map set-metric out
SWC (config-router-bgp)#neighbor 1.1.1.2 remote-as 300
SWC (config-router-bgp)#exit
SWC (config)#route-map set-metric permit 10
SWC (Config-Router-RouteMap)#set metric 120
```

SWD 的配置

```
SWD (config)#router bgp 300
```

```
SWD (config-router-bgp)#neighbor 3.3.3.3 remote-as 100
SWD (config-router-bgp)#neighbor 3.3.3.3 route-map set-metric out
SWD (config-router-bgp)#neighbor 1.1.1.1 remote-as 300
SWD (config-router-bgp)#exit
SWD (config)#route-map set-metric permit 10
SWD (Config-Router-RouteMap)#set metric 200
```

SWB 的配置

```
SWB (config)#router bgp 400
SWB (config-router-bgp)#neighbor 4.4.4.4 remote-as 100
SWB (config-router-bgp)#neighbor 4.4.4.4 route-map set-metric out
SWB (config-router-bgp)#exit
SWB (config)#route-map set-metric permit 10
SWB (Config-Router-RouteMap)#set metric 50
```

经过上面的配置以后，假设 SWB、SWC 和 SWD 都向 SWA 发送了一条路由 12.0.0.0。根据 BGP 路由策略的比较，假设在进行 metric 属性比较之前，从上述三个交换机发出的该条路由的所有属性都相同。此时，metric 值较小的将是更优路径。但交换机对 metric 属性的比较将只在来自同一 AS 的路由上进行比较。则此时对于 SWA 来说，通过 SWC 的路径优于通过 SWD 的路径。而由于此时 SWC 和 SWB 位于不同的 AS，所以 SWA 将不在两个交换机之间进行 metric 值。如果要使 metric 值的比较在不同的 AS 之间也进行那么需要用到“bgp always-compare-med”命令。如果配置了该命令，则对于 SWA 来说通过 SWB 的路径是最优的。此时可以在 SWA 上增加下面的配置命令：

```
SWA(config-router-bgp)# bgp always-compare-med
```

7.3.7 案例七：BGP VPN典型案例

在 MPLS VPN 环境下，BGP 作为核心的路由传播工具和在边缘设备上产生 ILM 和 FTN 表项的重要工具，在系统上，它和 LDP 协议一起，为 MPLS VPN 的应用提供了核心支持。在这里，LDP 工作在公网上，而在公网的非边缘路由器上，不需要运行 BGP 协议。

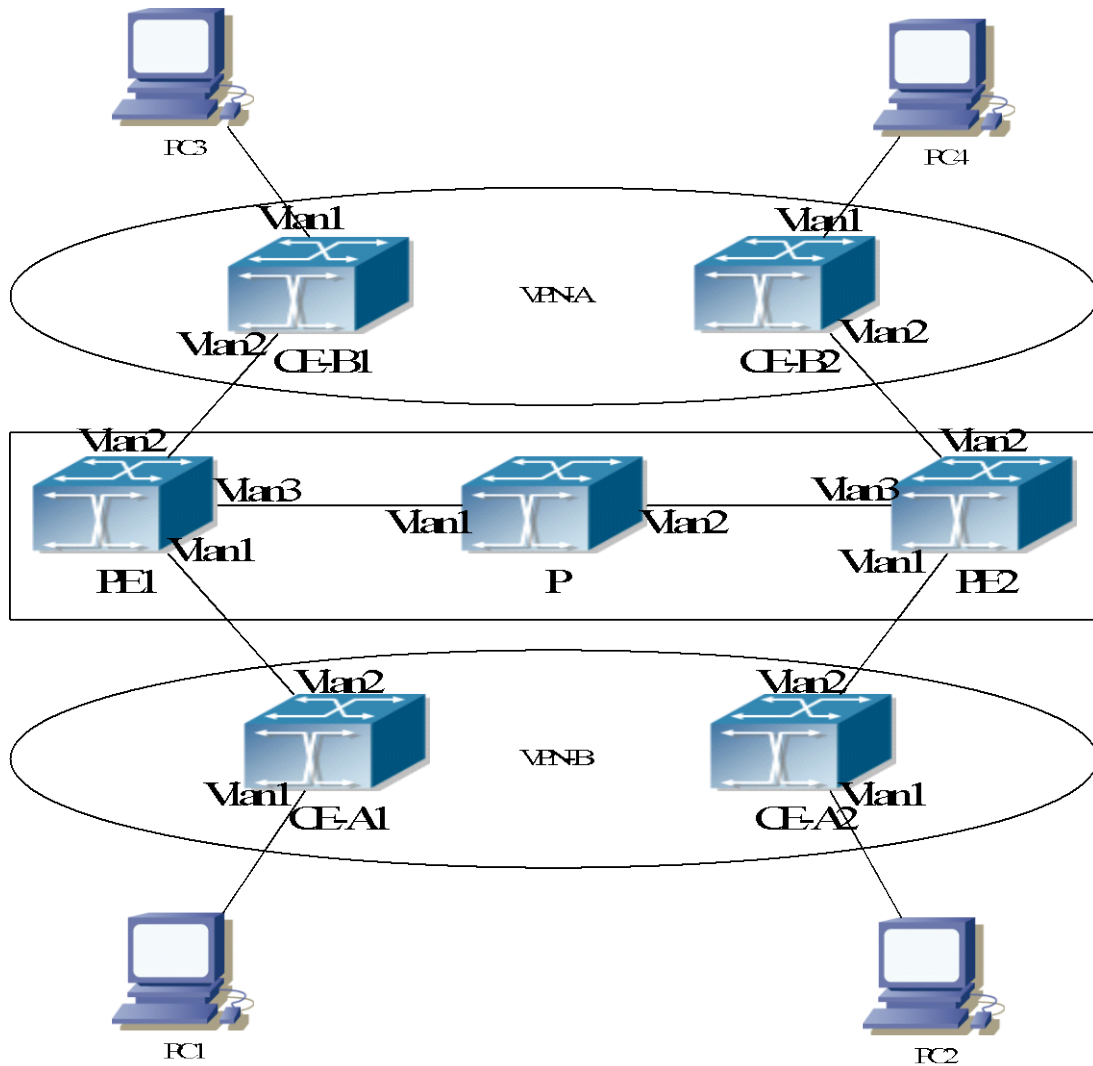


图 7-5 MPLS VPN 典型案例

如图是一个典型的 MPLS VPN 案例，其中 PE1、P、PE2 组成了公网区域，也就是使用 MPLS 交换的区域，CE-A1 与 CE-A2 组成了 VPN-A，CE-B1 与 CE-B2 组成了 VPN-B。两个 VPN 都通过公网的标签交换通信，他们互相不能通信。PE1 和 PE2 是运营商提供的边缘路由器，而 CE-A1、CE-A2、CE-B1、CE-B2 是用户端的接入交换机，PC1-PC4 表示了接入的用户。BGP 分别运行在公网区域和私网区域，其中在公网区域需要支持 VPN 路由，并且使用 LOOPBACK 接口进行连接。

各设备配置如下：

CE-A1 的配置：

```
CE-A1#config
```

```
CE-A1(config)#interface vlan 2
```

```
CE-A1(config-if-Vlan2)#ip address 192.168.101.2 255.255.255.0
```

```
CE-A1(config-if-Vlan2)#exit
```

```
CE-A1(config)#interface vlan 1
```

```
CE-A1(config-if-Vlan2)#ip address 10.1.1.1 255.255.255.0
```

```
CE-A1(config-if-Vlan2)#exit
CE-A1(config)#router bgp 60101
CE-A1(config-router)#neighbor 192.168.101.1 remote-as 100
CE-A1(config-router)#exit
```

CE-A2 的配置:

```
CE-A2#config
CE-A2(config)#interface vlan 2
CE-A2(config-if-Vlan2)#ip address 192.168.102.2 255.255.255.0
CE-A2(config-if-Vlan2)#exit
CE-A2(config)#interface vlan 1
CE-A2(config-if-Vlan2)#ip address 10.1.2.1 255.255.255.0
CE-A2(config-if-Vlan2)#exit
CE-A2(config)#router bgp 60102
CE-A2(config-router)#neighbor 192.168.102.1 remote-as 100
CE-A2(config-router)#exit
```

CE-B1 的配置:

```
CE-B1#config
CE-B1(config)#interface vlan 2
CE-B1(config-if-Vlan2)#ip address 192.168.201.2 255.255.255.0
CE-B1(config-if-Vlan2)#exit
CE-B1(config)#interface vlan 1
CE-B1(config-if-Vlan2)#ip address 20.1.1.1 255.255.255.0
CE-B1(config-if-Vlan2)#exit
CE-B1(config)#router bgp 60201
CE-B1(config-router)#neighbor 192.168.201.1 remote-as 100
CE-B1(config-router)#exit
```

CE-B2 的配置:

```
CE-B2#config
CE-B2(config)#interface vlan 2
CE-B2(config-if-Vlan2)#ip address 192.168.202.2 255.255.255.0
CE-B2(config-if-Vlan2)#exit
CE-B2(config)#interface vlan 1
CE-B2(config-if-Vlan2)#ip address 20.1.2.1 255.255.255.0
CE-B2(config-if-Vlan2)#exit
CE-B2(config)#router bgp 60202
```

```
CE-B2(config-router)#neighbor 192.168.202.1 remote-as 100
CE-B2(config-router)#exit
```

PE1 的配置:

```
PE1#config
PE1(config)#ip vrf VRF-A
PE1(config-vrf)#rd 100:10
PE1(config-vrf)#route-target both 100:10
PE1(config-vrf)#exit
PE1(config)#ip vrf VRF-B
PE1(config-vrf)#rd 100:20
PE1(config-vrf)#route-target both 100:20
PE1(config-vrf)#exit
PE1(config)#interface vlan 1
PE1(config-if-Vlan1)#ip vrf forwarding VRF-A
PE1(config-if-Vlan1)#ip address 192.168.101.1 255.255.255.0
PE1(config-if-Vlan1)#exit
PE1(config)#interface vlan 2
PE1(config-if-Vlan2)#ip vrf forwarding VRF-B
PE1(config-if-Vlan2)#ip address 192.168.201.1 255.255.255.0
PE1(config-if-Vlan2)#exit
PE1(config)#interface vlan 3
PE1(config-if-Vlan3)#ip address 202.200.1.2 255.255.255.0
PE1(config-if-Vlan3)#label-switching
PE1(config-if-Vlan3)#exit
PE1(config)#interface loopback 1
PE1(Config-if-Loopback1)# ip address 200.200.1.1 255.255.255.255
PE1(config-if-Vlan3)#exit
PE1(config)#router bgp 100
PE1(config-router)#neighbor 200.200.1.2 remote-as 100
PE1(config-router)#neighbor 200.200.1.2 update-source 200.200.1.1
PE1(config-router)#address-family vpnv4 unicast
PE1(config-router-af)#neighbor 200.200.1.2 activate
PE1(config-router-af)#exit-address-family
PE1(config-router)#address-family ipv4 vrf VRF-A
PE1(config-router-af)# neighbor 192.168.101.2 remote-as 60101
PE1(config-router-af)#exit-address-family
PE1(config-router)#address-family ipv4 vrf VRF-B
```

```
PE1(config-router-af)# neighbor 192.168.201.2 remote-as 60201
```

```
PE1(config-router-af)#exit-address-family
```

PE2 的配置:

```
PE2#config
```

```
PE2(config)#ip vrf VRF-A
```

```
PE2(config-vrf)#rd 100:10
```

```
PE2(config-vrf)#route-target both 100:10
```

```
PE2(config-vrf)#exit
```

```
PE2(config)#ip vrf VRF-B
```

```
PE2(config-vrf)#rd 100:20
```

```
PE2(config-vrf)#route-target both 100:20
```

```
PE2(config-vrf)#exit
```

```
PE2(config)#interface vlan 1
```

```
PE2(config-if-Vlan1)#ip vrf forwarding VRF-A
```

```
PE2(config-if-Vlan1)#ip address 192.168.102.1 255.255.255.0
```

```
PE2(config-if-Vlan1)#exit
```

```
PE2(config)#interface vlan 2
```

```
PE2(config-if-Vlan2)#ip vrf forwarding VRF-B
```

```
PE2(config-if-Vlan2)#ip address 192.168.202.1 255.255.255.0
```

```
PE2(config-if-Vlan2)#exit
```

```
PE2(config)#interface vlan 3
```

```
PE2(config-if-Vlan3)#ip address 202.200.2.2 255.255.255.0
```

```
PE2(config-if-Vlan3)#label-switching
```

```
PE2(config-if-Vlan3)#exit
```

```
PE2(config)#interface loopback 1
```

```
PE2(Config-if-Loopback1)# ip address 200.200.1.2 255.255.255.255
```

```
PE2(config-if-Vlan3)#exit
```

```
PE2(config)#router bgp 100
```

```
PE2(config-router)#neighbor 200.200.1.1 remote-as 100
```

```
PE2(config-router)#address-family vpnv4 unicast
```

```
PE2(config-router-af)#neighbor 200.200.1.1 activate
```

```
PE2(config-router-af)#exit-address-family
```

```
PE2(config-router)#address-family ipv4 vrf VRF-A
```

```
PE2(config-router-af)# neighbor 192.168.102.2 remote-as 60102
```

```
PE2(config-router-af)#exit-address-family
```

```
PE2(config-router)#address-family ipv4 vrf VRF-B
```

```
PE2(config-router-af)# neighbor 192.168.202.2 remote-as 60202
```

PE2(config-router-af)#exit-address-family

上面的例子是一种比较典型的情况，如果需要 VRF 间可以通信，可以通过修改 route-target 的方式实现。如果两端的 BGP AS 号相同，还需要在 PE 上使用 neighbor <ip-addr> as-override 命令来避免 AS 号的重复。

这里只列出了 BGP 相关的配置，公网上运行的 LDP 协议的相关配置请参考 LDP 典型案例。

7.4 BGP排错帮助

在配置、使用 BGP 协议时，可能会由于物理连接、配置错误等原因导致 BGP 协议未能正常运行。因此，用户应注意以下要点：

- ☞ 首先应该保证物理连接的正确无误。
- ☞ 其次，保证接口和链路协议是 UP（使用 show interface 命令）。
- ☞ 然后，先启动 BGP 协议（使用 router bgp 命令），再配置相应的 IBGP 和 EBGP 邻居（使用 neighbor remote-as 命令）。
- ☞ 需注意 BGP 协议本身并不能发现路由，BGP 路由的产生需要引入其他的路由。只有引入了路由才能将这些路由通告到 IBGP 和 EBGP 邻居上。这些引入的路由包括直连路由、静态路由、IGP 路由（RIP 和 OSPF）。引入这些路由的方法可以通过 network 命令和 redistribute（BGP）命令。
- ☞ 对于 BGP 来说，IBGP 和 EBGP 的行为不同需要注意。
- ☞ 当配置完成后，可以使用 show ip bgp summary 命令察看邻居的连接情况，应确保所有的邻居都保持 BGP 连接状态。并使用 show ip bgp 命令察看 BGP 路由表，使用 show ip route 察看 IP 路由表。
- ☞ 如果仍无法解决 BGP 的路由问题，那么请使用 debug ip bgp 等调试命令，然后将 3 分钟内的 debug 信息拷贝下来，发送给本公司技术服务中心。

第8章 MBGP4+

8.1 MBGP4+简介

MBGP4+是多协议BGP(Multiprotocol Border Gateway Protocol)为IPv6进行的扩展，关于BGP协议的介绍请参看本手册的BGP协议部分。与RIPng和OSPFv3不同，由于BGP本身的灵活性，BGP对IPv6没有单独的协议，而是在原有的BGP上进行了地址族的扩展，MBGP4+对BGP的扩展主要体现在：

- a. 配置的邻居地址可以是IPv6地址；
- b. 增加IPv6 unicast 地址族的配置。

8.2 MBGP4+配置

MBGP4+配置任务序列：

1. 配置IPv6邻居
2. 配置使能IPv6地址族
3. 配置MBGP4+引入不同进程OSPFv3路由
 - (1) 启动MBGP4+引入不同进程OSPFv3路由功能
 - (2) 显示和调试MBGP4+引入不同进程OSPFv3路由功能相关信息

1. 配置IPv6邻居

命令	解释
BGP 协议配置模式	
neighbor <X:X::X:X> remote-as <as-id>	配置 IPv6 邻居。

2. 配置使能IPv6地址族

命令	解释
BGP 协议配置模式	
address-family IPv6 unicast	进入 IPv6 单播地址族。
BGP 协议地址族配置模式	
neighbor <X:X::X:X> activate no neighbor <X:X::X:X> activate	配置 IPv6 邻居使能/不使能该地址族。
exit-address-family	退出地址族配置模式。

3. 配置MBGP4+引入不同进程OSPFv3路由

- (1) 启动MBGP4+引入不同进程OSPFv3路由功能

命令	解释
Router ipv6 bgp 配置模式	
redistribute ospf [<process-tag>] [route-map<word>] no redistribute ospf [<process-tag>]	启动或者关闭 MBGP4+引入其他 OSPFv3 进程路由功能。

(2) 显示和调试 MBGP4+引入不同进程 OSPFv3 路由功能相关信息

命令	解释
特权和配置模式	
show ipv6 bgp redistribute	显示 bgp4+进程引入其它外部路由的配置信息。
特权模式	
debug ipv6 bgp redistribute message send no debug ipv6 bgp redistribute message send debug ipv6 bgp redistribute route receive no debug ipv6 bgp redistribute route receive	打开或关闭 MBGP4+引入其他 OSPFv3 进程路由的命令下发调试开关。 打开或关闭 MBGP4+进程收到 nsm 发来的路由信息的调试开关。

8.3 MBGP4+案例

SwitchB、SwitchC和SwitchD位于AS200中，SwitchA位于AS100中。SwitchA和SwitchB共享一个相同的网络段2001::/96。而SwitchB和SwitchD彼此物理上不相邻。

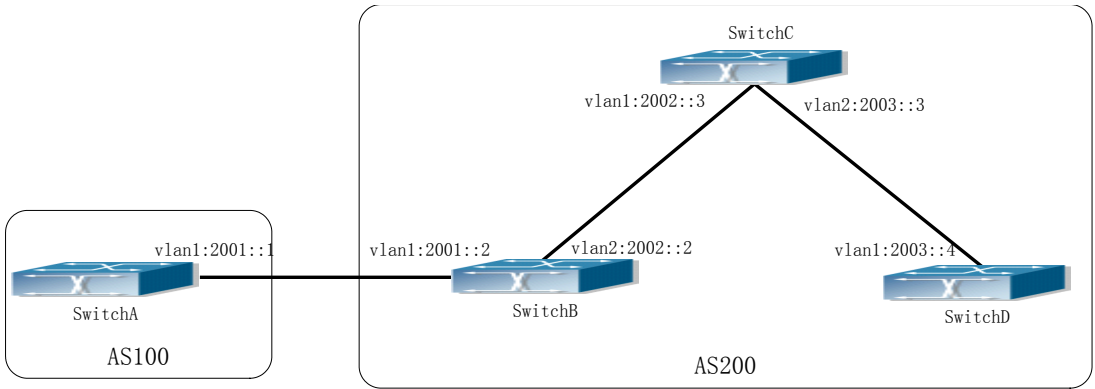


图 8-1 BGP 网络拓扑图

则SwitchA的配置如下：

```
SwitchA(config)#router bgp 100
SwitchA(config-router)#bgp router-id 1.1.1.1
SwitchA(config-router)#neighbor 2001::2 remote-as 200
SwitchA(config-router)#address-family IPv6 unicast
SwitchA(config-router-af)#neighbor 2001::2 activate
SwitchA(config-router-af)#exit-address-family
SwitchA(config-router)#exit
SwitchA(config)#
```

SwitchB的配置如下：

```
SwitchB(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
SwitchB(config-router)#neighbor 2001::1 remote-as 100
SwitchB(config-router)#neighbor 2002::3 remote-as 200
SwitchB(config-router)#neighbor 2003::4 remote-as 200
SwitchB(config-router)#address-family IPv6 unicast
SwitchB(config-router-af)#neighbor 2001::1 activate
SwitchB(config-router-af)#neighbor 2002::3 activate
SwitchB(config-router-af)#neighbor 2003::4 activate
SwitchB(config-router-af)#exit-address-family
SwitchB(config-router)#exit
SwitchB(config)#
```

SwitchC的配置如下：

```
SwitchC(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
SwitchC(config-router)#neighbor 2002::2 remote-as 200
SwitchC(config-router)#neighbor 2003::4 remote-as 200
SwitchC(config-router)#address-family IPv6 unicast
SwitchC(config-router-af)#neighbor 2002::2 activate
SwitchC(config-router-af)#neighbor 2003::4 activate
SwitchC(config-router-af)#exit-address-family
SwitchC(config-router)#exit
```

RouteD的配置如下：

```
SwitchD(config)#router bgp 200
SwitchB(config-router)#bgp router-id 2.2.2.2
```

```
SwitchD(config-router)#neighbor 2003::3 remote-as 200
SwitchD(config-router)#neighbor 2002::2 remote-as 200
SwitchD(config-router)#address-family IPv6 unicast
SwitchD(config-router-af)#neighbor 2002::2 activate
SwitchD(config-router-af)#neighbor 2003::3 activate
SwitchD(config-router-af)#exit-address-family
SwitchD(config-router)#exit
```

此时SwitchB和SwitchA之间的连接是EBGP，和SwitchC、SwitchD之间的连接是IBGP。

SwitchB和SwitchD之间不必物理上相连就可以进行BGP的连接，但前提是两个交换机必须有到达对方的路由。该路由可以通过静态路由或是IGP获得。

8.4 MBGP4+排错帮助

同 7.4 节 BGP 排错帮助。

第9章 黑洞路由操作手册

9.1 黑洞路由介绍

黑洞路由实际是一种特殊的静态路由，顾名思义，就是目的地址为该网段的数据报文到达设备之后，将被丢弃。

9.2 IPv4 黑洞路由配置任务

1. 配置IPv4黑洞路由

1. 配置IPv4黑洞路由

命令	解释
全局配置模式	
ip route {<ip-prefix> <mask> <ip-prefix> <prefix-length>} null0 [<distance>] no ip route {<ip-prefix> <mask> <ip-prefix> <prefix-length>} null0	配置静态黑洞路由。本命令的no操作作为删除配置的黑洞路由。

9.3 IPv6 黑洞路由配置任务

1. 启动IPv6功能
2. 配置IPv6黑洞路由

1. 启动IPv6功能

命令	解释
全局配置模式	
ipv6 enable	在交换机上启动IPv6功能。

2. 配置IPv6黑洞路由

命令	解释
全局配置模式	
ipv6 route <ipv6-prefix prefix-length> null0 [<precedence>] no ipv6 route <ipv6-prefix prefix-length> null0	配置IPv6静态黑洞路由。本命令的no操作作为删除配置的IPv6黑洞路由。

9.4 黑洞路由举例

案例1: IPv4黑洞路由功能。

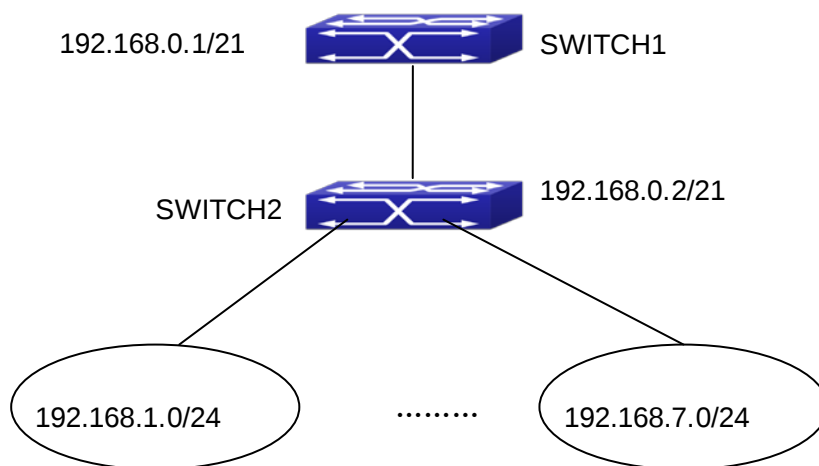


图9-1配置IPv4黑洞路由的示例图

如图所示，交换机 SWITCH2 配置了 8 个 VLAN 三层接口用于用户接入，分别是 192.168.1.0/24 ~ 192.168.7.0/24，同时 SWITCH2 配置了一条缺省路由指向路由器 SWITCH1，SWITCH1 配置回程路由 192.168.0.0/21 指回到 S。正常情况下这个网络不会出现问题，但是 SWITCH2 的一个三层接口 down 掉之后，例如到网段 192.168.1.0/24 的三层接口 down 掉了，有数据包发往某 VLAN1 时，但数据到达 SWITCH2 之后，没有到这个网段的路由，那么会根据缺省路由送到 SWITCH1，SWITCH1 在接受到 IP 报文之后，查询路由表，匹配上直连路由，然后又送到 SWITCH2，如此往复，就在 SWITCH2 和 SWITCH1 之间循环转发，形成路由环路。为了解决这个问题，可以在 SWITCH2 上配置一条“黑洞路由”：

```
ip route 192.168.0.0/21 null0 50
```

这时，当到 SWITCH2 收到 interface vlan1 报文之后，查询路由表，匹配上这条黑洞路由，那么会丢弃该报文，便不会在 SWITCH2 和 SWITCH1 上循环转发。

配置步骤如下：

```
Switch#config
```

```
Switch(config)#ip route 192.168.0.0/21 null0 50
```

案例2: IPv6黑洞路由功能。

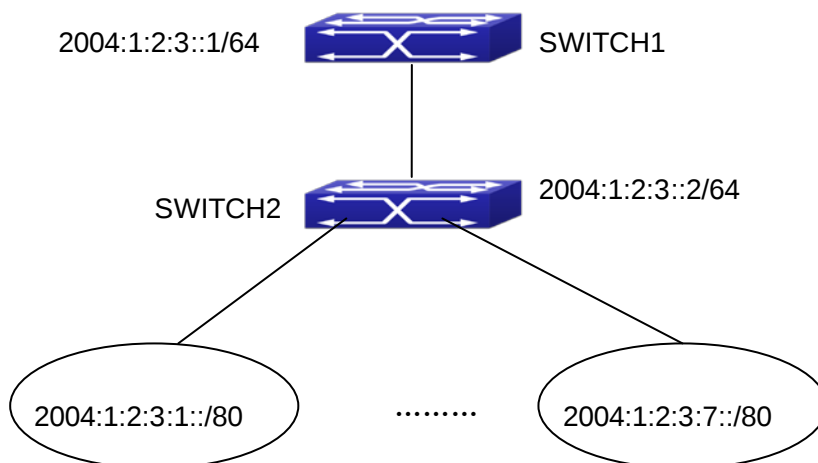


图9-2配置IPv6黑洞路由的示例图

如图所示，交换机 SWITCH2 配置了 8 个 VLAN 三层接口用于用户接入，分别是 2004:1:2:3:1/80 ~ 2004:1:2:3:7/80，同时 SWITCH2 配置了一条缺省路由指向路由器 SWITCH1，SWITCH1 配置回程路由 2004:1:2:3::/64 指回到 SWITCH2。正常情况下这个网络不会出现问题，但是 SWITCH2 的一个三层接口 down 掉之后，例如到网段 2004:1:2:3:1/80 的三层接口 down 掉了，有数据包发往某 VLAN1 时，但数据到达 SWITCH2 之后，没有到这个网段的路由，那么会根据缺省路由送到 SWITCH1，SWITCH1 在接受到 IP 报文之后，查询路由表，匹配上直连路由，然后又送到 SWITCH2，如此往复，就在 SWITCH2 和 SWITCH1 之间循环转发，形成路由环路。为了解决这个问题，可以在 SWITCH2 上配置一条“黑洞路由”：

```
ipv6 route 2004:1:2:3::/64 null0 50
```

这时，当到 SWITCH2 收到 interface vlan1 报文之后，查询路由表，匹配上这条黑洞路由，那么会丢弃该报文，便不会在 SWITCH2 和 SWITCH1 上循环转发。

配置步骤如下：

```
Switch#config
```

```
Switch(config)#ipv6 route 2004:1:2:3::/64 null0 50
```

9.5 黑洞路由排错帮助

在配置、使用黑洞路由功能时，可能会因为掩码长度不合适，以及管理距离不合适而造成无法正常使用。因此，用户应注意以下要点：

- ☞ 使用 IPv6 黑洞路由时，请首先确保全局启动了 IPv6 功能。
- ☞ 掩码长度最好比正常的路由的掩码长度长，这样可以确保当掩码长度长的正常路由有效的时候不干扰其工作。
- ☞ 当掩码长度与其他路由一样时，建议把黑洞路由的 distance 调的低一点。

如果使用检查都尝试仍无法解决的问题，那么请使用 `show ip route database`，以及 `show ip route fib`，`show I3` 等命令，然后将这些信息拷贝下来，发送给本公司技术服务中心。

第10章 GRE隧道配置

10.1 GRE隧道介绍

GRE(通用路由协议封装)是由 Cisco 和 Net-smiths 等公司于 1994 年提交给 IETF 的,标号为 RFC1701 和 RFC1702。目前有多数厂商的网络设备均支持 GRE 隧道协议。GRE 规定了如何用一种网络协议去封装另一种网络协议的方法。GRE 的隧道由两端的源 IP 地址和目的 IP 地址来定义,允许用户使用 IP 包封装 IP、IPX、AppleTalk 包,并支持全部的路由协议(如 RIP2、OSPF 等)。通过 GRE,用户可以利用公共 IP 网络连接 IPX 网络、AppleTalk 网络,还可以使用保留地址进行网络互连,或者对公网隐藏企业网的 IP 地址。GRE 只提供了数据包的封装,它并没有加密功能来防止网络侦听和攻击。所以在实际环境中它常和 IPsec 在一起使用,由 IPsec 提供用户数据的加密,从而给用户提供更好的安全性。

GRE 协议的主要用途有两个:企业内部协议封装和私有地址封装。在国内,由于企业网几乎全部采用的是 TCP/IP 协议,因此在中国建立隧道时没有对企业内部协议封装的市场需求。企业使用 GRE 的唯一理由应该是对内部地址的封装。我们交换机应用 GRE 主要是因为互联网协议的过渡(包括 IPv6 OVER IPv4 和 IPv4 OVER IPv6)。

本文工作根据 RFC1701、1702、2784 实现。

10.2 GRE隧道基本配置

GRE 隧道配置任务序列如下:

1. 配置隧道模式
 - 1) 配置隧道模式为 GREv4 隧道
 - 2) 配置隧道模式为 GREv6 隧道
2. 配置 GRE 隧道的源地址和目的地址
 - 1) 配置 GRE 隧道的源地址为 IPv6 或者 IPv4 地址
 - 2) 配置 GRE 隧道的目的地址为 IPv6 或者 IPv4 地址
3. 配置 GRE 隧道接口地址
 - 1) 配置 GRE 隧道接口的 IPv4 地址
 - 2) 配置 GRE 隧道接口的 IPv6 地址
4. 配置静态路由的出接口是 GRE 隧道
 - 1) 配置 IPv4 静态路由的出接口是 GRE 隧道
 - 2) 配置 IPv6 静态路由的出接口是 GRE 隧道

1. 配置隧道模式

命令	解释
隧道接口配置模式	
tunnel mode gre ip no tunnel mode	配置隧道模式为 GREv4 隧道，数据报文 GRE 封装后是 IPv4 报文头，穿越 IPv4 网络。
tunnel mode gre ipv6 no tunnel mode	配置隧道模式为 GREv6 隧道，数据报文 GRE 封装后是 IPv6 报文头，穿越 IPv6 网络。

2. 配置 GRE 隧道的源地址和目的地址

命令	解释
隧道接口配置模式	
tunnel source {<ipv6-address> <ipv4-address>} no tunnel source	配置 GRE 隧道的源地址为 IPv6 或者 IPv4 地址。
tunnel destination {<ipv6-address> <ipv4-address>} no tunnel destination	配置 GRE 隧道的目的地址为 IPv6 或者 IPv4 地址。

3. 配置 GRE 隧道接口地址

命令	解释
隧道接口配置模式	
ip address <ipv4-address> <mask> no ip address <ipv4-address> <mask>	配置 GRE 隧道接口的 IPv4 地址。
ipv6 address <ipv6-address/prefix> no ipv6 address <ipv6-address/prefix>	配置 GRE 隧道接口的 IPv6 地址。

4. 配置静态路由的出接口是 GRE 隧道

命令	解释
全局配置模式	
ip route <ipv4-address/mask> tunnel <ID> no ip route <ipv4-address/mask> tunnel <ID>	配置 IPv4 静态路由的出接口是 GRE 隧道。
ipv6 route <ipv6-address/prefix> tunnel <ID> no ipv6 route <ipv6-address/prefix> tunnel <ID>	配置 IPv6 静态路由的出接口是 GRE 隧道。

10.3 GRE隧道举例

GRE 隧道典型配置举例：

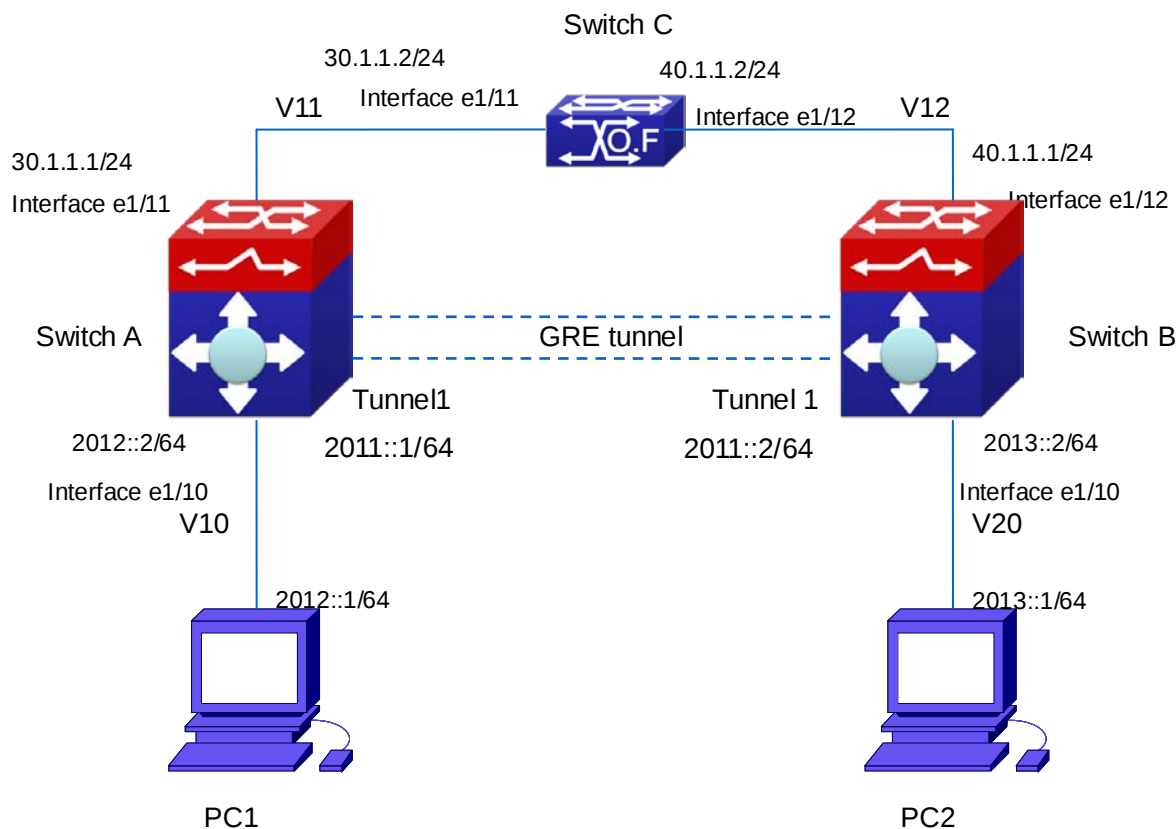


图 10-1 IPv6 over IPv4 GRE 隧道特性典型配置组网图

配置思路

- ☞ 配置 IPv4 网络，保证 IPv4 互通。
- ☞ 配置设备的隧道接口、连接 PC 的业务接口。
- ☞ 配置隧道参数，使能隧道接口。
- ☞ 使能 OSPF 路由协议使得 PC1 和 PC2 间数据通过隧道转发。

配置步骤

说明：本文的组网环境可能与您的实际环境存在差异。为了保证配置效果，请确认设备上现有配置和以下配置不冲突。

(1) 设备 A 的配置：

1. 配置步骤

- ☞ 创建业务 VLAN 11 及其接口地址。

```
SwitchA(config)#Vlan 11
```

```
SwitchA(config-vlan11)#switchport interface ethernet 1/11
```

```
SwitchA(config-vlan11)#exit
```

```
SwitchA(config)#interface vlan 11
```

```
SwitchA(config-if-vlan11)#ip address 30.1.1.1/24
```

☞ 配置从 SwitchA 到 SwitchB 上接口 Vlan11 的 IPv4 静态路由。

```
SwitchA(config)#ip route 40.1.1.1/24 30.1.1.2
```

☞ 配置隧道接口及类型和源端、目的端。当隧道起来以后，隧道的源和目的不允许修改，除非隧道源为三层接口时，只有在更改三层接口地址的时候允许更新隧道源

```
SwitchA(config)#interface tunnel 1
```

```
SwitchA(config-if-tunnel1)# tunnel source 30.1.1.1
```

```
SwitchA(config-if-tunnel1)# tunnel destination 40.1.1.1
```

```
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ip	30.1.1.1	40.1.1.1

可以看到 GRE 隧道已经配置成功。

☞ 配置隧道接口 IPv6 地址。对于隧道接口配置 IPv4/IPv6 地址只允许配置一个接口地址，该限制同样适用于其它模式的隧道，如配置隧道、6to4、isatap 等。

注意：配置 IPv4 地址的时候由于实现的原因必须要求隧道 active，存在配置保留后丢失 IPv4 地址配置的可能，这个和 IPv6 地址的配置逻辑不一致，IPv6 不要求隧道必须 active。

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::1/64
```

☞ 配置业务 VLAN10 及其接口地址。

```
SwitchA(config)#vlan 10
```

```
SwitchA(config-vlan10)#switchport interface ethernet 1/10
```

```
SwitchA(config-vlan10)#exit
```

```
SwitchA(config)#interface vlan 10
```

```
SwitchA(config-if-vlan10)# ipv6 address 2012::2/64
```

```
SwitchA(config-if-vlan10)#exit
```

☞ 配置 OSPF 路由协议。

```
SwitchA(config)#router ospf
```

```
SwitchA(config-router)#router-id 1.1.1.1
```

```
SwitchA(config-router)#network 30.1.1.1/24 area 0
```

```
SwitchA(config-router)#exit
```

(2) 设备 B 的配置：

1. 配置步骤

☞ 创建业务 VLAN 12 及其接口地址。

```
SwitchA(config)#vlan 12
```

```
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 40.1.1.1/24
SwitchA(config-if-vlan12)#exit
SwitchA(config)#
```

- ☞ 配置从 SwitchB 到 SwitchA 上接口 Vlan12 的 IPv4 静态路由。

```
SwitchA(config)#ip route 30.1.1.1/24 40.1.1.2
```

- ☞ 配置隧道接口及类型和源端、目的端。

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 2005:3000:1000::1
SwitchA(config-if-tunnel1)# tunnel destination 2000:3000:1000::1
SwitchA(config-if-tunnel1)# tunnel mode gre ipv6
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ipv6	2005:1000:3000::1	2000:1000:3000::1

可以看到 GRE 隧道已经配置成功。

- ☞ 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 OSPF 路由协议。

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::2/64
```

- ☞ 配置业务 VLAN20 及其接口地址。

```
SwitchA(config)#vlan 20
SwitchA(config-vlan20)#switchport interface ethernet 1/10
SwitchA(config-vlan20)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-vlan20)# ipv6 address 2013::2/64
SwitchA(config-if-vlan20)#exit
SwitchA(config)#
```

- ☞ 配置 OSPF 路由协议。

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.2
SwitchA(config-router)#network 40.1.1.0/24 area 0
SwitchA(config-router)#exit
SwitchA(config)#
```

(3) 设备 C 的配置

1. 配置步骤

- ☞ 创建业务 VLAN 11 及其接口地址。

```
SwitchA(config)#vlan 11
```

```
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.2/24
```

☞ 创建业务 VLAN 12 及其接口地址。

```
SwitchA(config)#vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan12)#ip address 40.1.1.2/24
SwitchA(config-if-vlan12)#exit
```

(4) PC 的配置

☞ 配置 PC1 的网卡地址及默认网关。

PC1: 网卡地址: 2012::1/64, 默认网关: 2012::2

PC2: 网卡地址: 2013::1/64, 默认网关: 2013::2

10.4 GRE隧道引用业务环回组举例

业务环回组功能简介:

若同一台设备存在多种业务单板混插的情况,需要通过业务环回功能来实现不同业务单板间的业务重定向。例如支持 GRE 隧道和不支持 GRE 隧道单板混插,此时就需要通过业务环回功能将不支持 GRE 隧道单板收到的 GRE 隧道数据环回到支持 GRE 隧道的单板上进行处理。业务环回组需要占用至少一个业务单板上的闲置端口,这些端口在加入环回组之前不能有任何配置。为了增加板间业务重定向的带宽,可以将多个端口加入到业务环回组中,这一组端口构成环回组,进行流量负载分担。

GRE 隧道引用业务环回组典型配置举例:

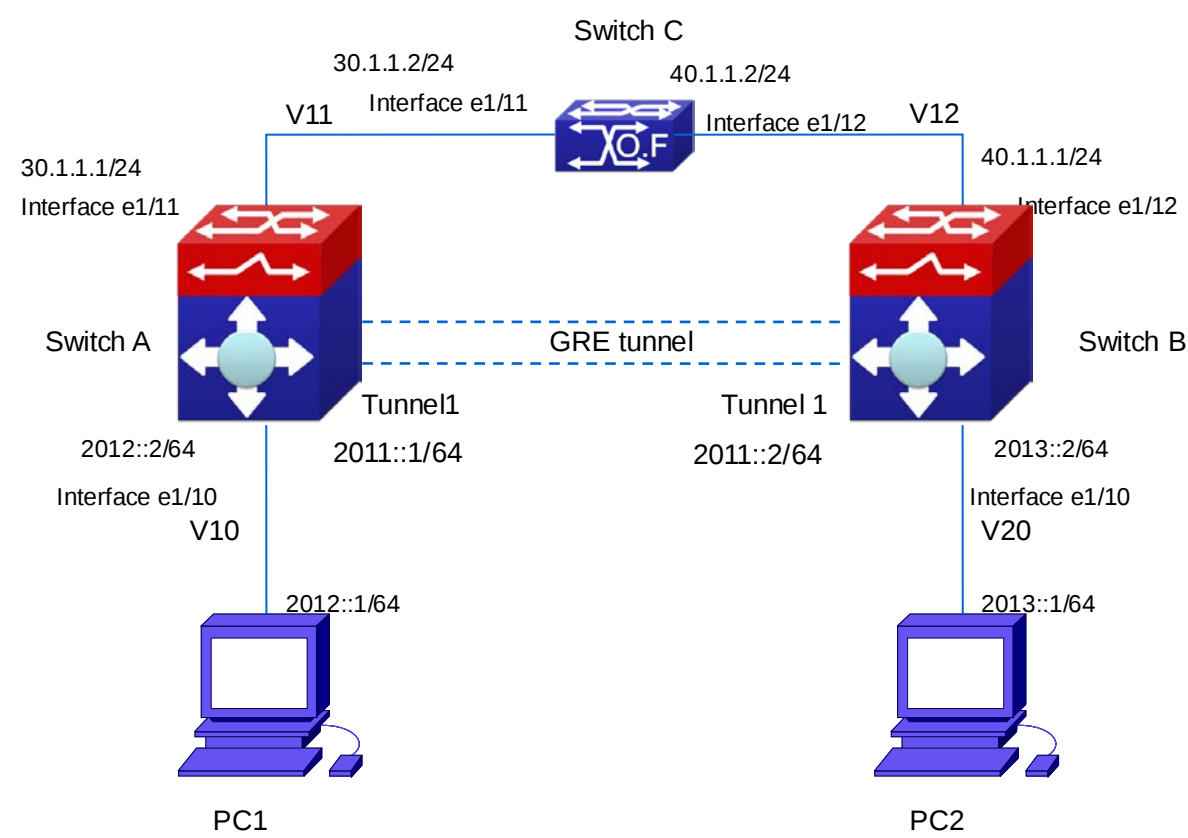


图 10-2 GRE 隧道引用业务环回组典型配置组网图

环回组组网图介绍

SwitchA 和 SwitchB 之间为 ipv4 网络，PC1 和 PC2 处在 IPv6 网络。如果 PC1 需要和远端的 PC2 通信，必须通过 GRE 隧道穿越 SwitchA 和 SwitchB 之间的 ipv4 网络。SwitchA 存在多种业务单板混插的情况：单板 1 不支持 GRE 隧道功能，单板 3 支持 GRE 隧道，因此单板 1 收到 PC1 发往 PC2 的数据需要通过业务环回组功能环回到单板 3 上进行处理

配置思路

- 配置 IPv4 网络，保证 IPv4 互通。
- 配置设备的隧道接口、连接 PC 的业务接口。
- 配置隧道参数，使能隧道接口。
- 配置环回组，把单板 3 上的闲置端口 1/12 加入此环回组并让隧道引用该环回组。
- 使能 OSPF 路由协议使得 PC1 和 PC2 间数据通过隧道转发。

配置步骤

说明：本文的组网环境可能与您的实际环境存在差异。为了保证配置效果，请确认设备上现有配置和以下配置不冲突。

（1）设备 A 的配置

1. 配置步骤

- 创建业务 VLAN 11 及其接口地址

```
SwitchA(config)#Vlan 11
SwitchA(config-vlan11)#switchport interface ethernet 1/11
SwitchA(config-vlan11)#exit
SwitchA(config)#interface vlan 11
SwitchA(config-if-vlan11)#ip address 30.1.1.1/24
```

- 配置从 SwitchA 到 SwitchB 上接口 Vlan11 的 IPv4 静态路由

```
SwitchA(config)#ip route 40.1.1.1/24 30.1.1.2
```

- 配置隧道接口及类型和源端、目的端

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ipv6	30.1.1.1	40.1.1.1

可以看到 GRE 隧道已经配置成功。

- 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 ospf 路由协议

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::1/64
```

- 配置业务 VLAN10 及其接口地址

```
SwitchA(config)#Vlan 10
SwitchA(config-vlan10)#switchport interface ethernet 1/10
SwitchA(config-vlan10)#exit
SwitchA(config)#interface vlan 10
SwitchA(config-if-vlan10)# ipv6 address 2012::2/64
SwitchA(config-if-vlan10)#exit
```

- 配置环回组并让隧道引用该环回组

```
SwitchA (config)#loopback-group 1
SwitchA (config-if-ethernet1/12)#loopback-group 1
SwitchA (config-if-tunnel1)# loopback-group 1
```

- 配置 OSPF 路由协议

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#network 30.1.1.0/24 area 0
SwitchA(config-router)#exit
```

(2) 设备 B 的配置

1. 配置步骤

- 创建业务 VLAN 12 及其接口地址

```
SwitchA(config)#Vlan 12
SwitchA(config-vlan12)#switchport interface ethernet 1/12
SwitchA(config-vlan12)#exit
SwitchA(config)#interface vlan 12
SwitchA(config-if-vlan11)#ip address 30.1.1.2/24
SwitchA(config-if-vlan12)#exit
SwitchA(config)#
```

- ☞ 配置从 SwitchB 到 SwitchA 上接口 Vlan12 的 IPv4 静态路由

```
SwitchA(config)#ip route 30.1.1.1/24 40.1.1.2
```

- ☞ 配置隧道接口及类型和源端、目的端

```
SwitchA(config)#interface tunnel 1
SwitchA(config-if-tunnel1)# tunnel source 40.1.1.1
SwitchA(config-if-tunnel1)# tunnel destination 30.1.1.1
SwitchA(config-if-tunnel1)# tunnel mode gre ip
```

```
SwitchA#show gre tunnel
```

name	mode	source	destination
Tunnel1	gre ipv6	40.1.1.1	30.1.1.1

可以看到 GRE 隧道已经配置成功。

- ☞ 配置隧道接口 IPv6 地址。接口地址一定要配，因为需要运行 ospf 路由协议

```
SwitchA (config-if-tunnel1)#ipv6 address 2011::2/64
```

- ☞ 配置业务 VLAN20 及其接口地址

```
SwitchA(config)#Vlan 20
SwitchA(config-vlan20)#switchport interface ethernet 1/10
SwitchA(config-vlan20)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-vlan20)# ipv6 address 2013::2/64
SwitchA(config-if-vlan20)#exit
SwitchA(config)#
```

- ☞ 配置 OSPF 路由协议

```
SwitchA(config)#router ospf
SwitchA(config-router)#router-id 1.1.1.2
SwitchA(config-router)#network 40.1.1.0/24 area 0
SwitchA(config-router)#exit
SwitchA(config)#
```

(3) 设备 C 的配置

1. 配置步骤

- ☞ 创建业务 VLAN 11 及其接口地址

```
SwitchA(config)#Vlan 11
```

```
SwitchA(config-vlan11)#switchport interface ethernet 1/11
```

```
SwitchA(config-vlan11)#exit
```

```
SwitchA(config)#interface vlan 11
```

```
SwitchA(config-if-vlan11)#ip address 30.1.1.2/24
```

☞ 创建业务 VLAN 12 及其接口地址

```
SwitchA(config)#vlan 12
```

```
SwitchA(config-vlan12)#switchport interface ethernet 1/12
```

```
SwitchA(config-vlan12)#exit
```

```
SwitchA(config)#interface vlan 12
```

```
SwitchA(config-if-vlan12)#ip address 40.1.1.2/24
```

```
SwitchA(config-if-vlan12)#exit
```

(4) PC 的配置

☞ 配置 PC1 的网卡地址及默认网关。

PC1: 网卡地址: 2012::1/64, 默认网关: 2012::2

PC2: 网卡地址: 2013::1/64, 默认网关: 2013::2

10.5 GRE隧道排错帮助

当在使用 GRE 隧道过程中遇到问题, 请检查是否是如下原因:

- ☞ 检查配置, 是否正确配置了隧道的源和目的地址, 配置隧道模式 (tunnel mode gre {ip | ipv6}) 是否正确。
- ☞ 在交换机配置了隧道后, 检查下一跳接口为 GRE 隧道接口的静态路由配置。
- ☞ 交换机之间的连线是否正常, 可通过 debug gre {packet | event | all} 观察交换机能否正确接收和处理相关的报文。

第11章 ECMP配置

11.1 ECMP简介

ECMP（Equal-cost Multi-path Routing）等价多路径路由，在存在多条不同链路到达同一目的地址的网络环境中，如果使用传统的路由技术，发往该目的地址的数据包只能利用其中的一条链路，其它链路处于备份状态或无效状态，并且在动态路由环境下相互的切换需要一定时间，而等价多路径路由协议可以在这样的网络环境下同时使用多条链路，不仅实现了流量分担，增加了传输带宽，并且可以无时延无丢包地备份失效链路的数据传输。

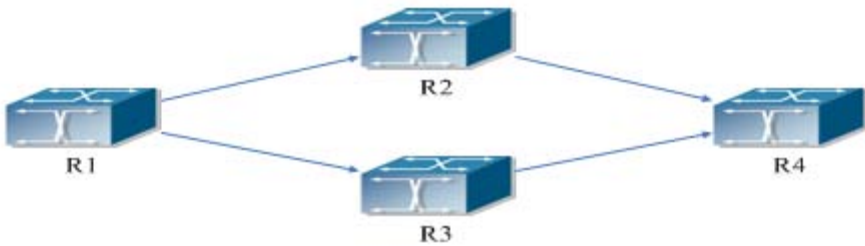


图 11-1 ECMP 的应用环境

如图所示，R1 到 R4 有两条路径可以选择，分别是 R1-R2-R4 和 R1-R3-R4，如果假设路由类型和代价开销相同，R1 到 R4 可以形成两条路由，目的地都是 R4，但是下一跳不相同，如果这两条路由都被选为最优，它们就形成了等价路由。

11.2 ECMP配置任务序列

1. 配置等价路由的最大数目

1. 配置等价路由的最大数目

命令	解释
全局配置模式	
maximum-paths <1-32> no maximum-paths	配置等价路由的最大数目。

11.3 ECMP典型案例

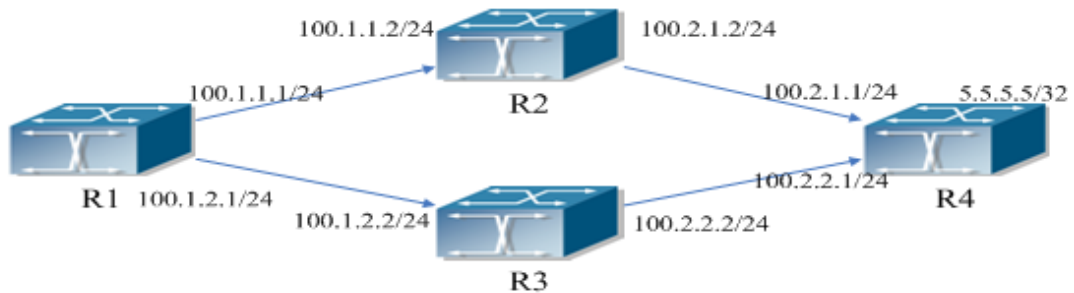


图 11-2 ECMP 的应用环境

如图所示，R1 与 R2、R3 连接的接口地址分别为 100.1.1.1/24、100.1.2.1/24。R2、R3 与 R1 连接的接口地址分别为 100.1.1.2/24、100.1.2.2/24。R4 与 R2、R3 连接的接口地址分别为 100.2.1.1/24、100.2.2.1/24。R2、R3 与 R4 连接的接口地址分别为 100.2.1.2/24、100.2.2.2/24，R4 的 loopback 地址为 5.5.5.5/32。

11.3.1 静态路由实现ECMP

```
R1(config)#ip route 5.5.5.5/32 100.1.1.2
```

```
R1(config)#ip route 5.5.5.5/32 100.1.2.2
```

在 R1 上显示路由表如下：

```
R1(config)#show ip route
```

Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP

O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

* - candidate default

```
C    1.1.1.1/32 is directly connected, Loopback1  tag:0
```

```
S    5.5.5.5/32 [1/0] via 100.1.1.2, Vlan100  tag:0
```

```
          [1/0] via 100.1.2.2, Vlan200  tag:0
```

```
C    100.1.1.0/24 is directly connected, Vlan100  tag:0
```

```
C    100.1.2.0/24 is directly connected, Vlan200 tag:0
```

```
C    127.0.0.0/8 is directly connected, Loopback  tag:0
```

Total routes are : 6 item(s)

11.3.2 OSPF实现ECMP

R1 的配置

```
R1(config)#interface Vlan100
R1(Config-if-Vlan100)# ip address 100.1.1.1 255.255.255.0
R1(config)#interface Vlan200
R1(Config-if-Vlan200)# ip address 100.1.2.1 255.255.255.0
R1(config)#interface loopback 1
R1(Config-if-loopback1)# ip address 1.1.1.1 255.255.255.255
R1(config)#router ospf 1
R1(config-router)# ospf router-id 1.1.1.1
R1(config-router)# network 100.1.1.0/24 area 0
R1(config-router)# network 100.1.2.0/24 area 0
```

R2 的配置

```
R2(config)#interface Vlan100
R2(Config-if-Vlan100)# ip address 100.1.1.2 255.255.255.0
R2(config)#interface Vlan200
R2(Config-if-Vlan200)# ip address 100.2.1.2 255.255.255.0
R2(config)#interface loopback 1
R2(Config-if-loopback1)# ip address 2.2.2.2 255.255.255.255
R2(config)#router ospf 1
R2(config-router)# ospf router-id 2.2.2.2
R2(config-router)# network 100.1.1.0/24 area 0
R2(config-router)# network 100.2.1.0/24 area 0
```

R3 的配置

```
R3(config)#interface Vlan100
R3(Config-if-Vlan100)# ip address 100.1.2.2 255.255.255.0
R3(config)#interface Vlan200
R3(Config-if-Vlan200)# ip address 100.2.2.2 255.255.255.0
R3(config)#interface loopback 1
R3(Config-if-loopback1)# ip address 3.3.3.3 255.255.255.255
R3(config)#router ospf 1
R3(config-router)# ospf router-id 3.3.3.3
R3(config-router)# network 100.1.2.0/24 area 0
R3(config-router)# network 100.2.2.0/24 area 0
```

R4 的配置

```
R4(config)#interface Vlan100
R4(Config-if-Vlan100)# ip address 100.2.1.1 255.255.255.0
R4(config)#interface Vlan200
R4(Config-if-Vlan200)# ip address 100.2.2.1 255.255.255.0
R4(config)#interface loopback 1
R4(Config-if-loopback1)# ip address 5.5.5.5 255.255.255.255
R4(config)#router ospf 1
R4(config-router)# ospf router-id 4.4.4.4
R4(config-router)# network 100.2.1.0/24 area 0
R4(config-router)# network 100.2.2.0/24 area 0
```

在 R1 上显示路由表如下：

```
R1(config)#show ip route
```

Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP

O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

* - candidate default

```
C    1.1.1.1/32 is directly connected, Loopback1  tag:0
O    5.5.5.5/32 [110/3] via 100.1.1.2, Vlan100, 00:00:05  tag:0
      [110/3] via 100.1.2.2, Vlan200, 00:00:05  tag:0
C    100.1.1.0/24 is directly connected, Vlan100  tag:0
C    100.1.2.0/24 is directly connected, Vlan200  tag:0
O    100.2.1.0/24 [110/2] via 100.1.1.2, Vlan100, 00:02:25  tag:0
O    100.2.2.0/24 [110/2] via 100.1.2.2, Vlan200, 00:02:25  tag:0
C    127.0.0.0/8 is directly connected, Loopback  tag:0
```

Total routes are : 8 item(s)

第12章 BFD

12.1 BFD介绍

BFD（Bidirectional Forwarding Detection，双向转发检测）是一套全网统一的检测机制，用于快速检测、监控网络中链路连通状况。为了提升现有网络性能，协议邻居之间必须能快速检测到通信故障，从而更快的建立起备用通道恢复通信。

BFD提供了一个通用的、标准化的、介质无关、协议无关的快速故障检测机制，可以为各上层协议如路由协议、MPLS 等统一地快速检测两台网络设备间双向转发路径的故障。BFD在两台网络设备上建立会话，用来监测两台网络设备间的双向转发路径，为上层协议服务。BFD本身并没有发现机制，而是靠被服务的上层协议通知其该与谁建立会话，会话建立后如果在检测时间内没有收到对端的BFD控制报文则认为发生故障，通知被服务的上层协议，上层协议进行相应的处理。

12.2 BFD配置任务序列

- 1. BFD 基本功能配置
- 2. BFD 功能与 RIP（ng）协议联动配置
- 3. BFD 功能与静态路由（IPv6）协议联动配置
- 4. BFD 功能与 VRRP（v3）协议联动配置

1. BFD 基本功能配置

命令	解释
全局配置模式	
bfd mode{active passive} no bfd mode	配置 BFD 会话建立前的工作模式，默认为 active 模式。本命令的 no 操作为恢复 active 模式。
bfd authentication key <1-255> text <WORD> no bfd authentication key <1-255>	配置 BFD 认证使用的 key 和文本方式加密的认证字符串。 no 命令为删除配置的 key。
bfd authentication key <1-255> md5 <WORD> no bfd authentication key	配置 BFD 使用的 key 和 md5 方式加密的认证字符串。no 命令为删除配置的 key。
接口配置模式	
bfd interval <value1> min_rx <value2> multiplier <value3>	配置 BFD 控制报文的最小发送时间间隔和会话检测系数。

no bfd interval	本命令的 no 操作为恢复检测系数为默认值。
bfd min-echo-receive-interval <value> no bfd min-echo-receive-interval	配置 BFD 控制报文的最小接收时间间隔, no 命令操作为恢复 echo 报文的最小接收间隔为默认值。
bfd echo no bfd echo	开启 bfd echo 功能, no 命令为关闭 bfd echo 功能。
bfd echo-source-ip <ipv4-address> no bfd echo-source-ip	配置的 Echo 报文源地址进行链路故障的探测, no 命令操作为删除配置的 echo 报文源地址。
bfd echo-source-ipv6 <ipv6-address> no bfd echo-source-ipv6	配置的 Echo 报文源地址进行链路故障的探测, no 命令操作为删除配置的 echo 报文源地址。
bfd authentication key <1-255> no bfd authentication key	接口开启 BFD 认证功能和配置使用的密钥。本命令的 no 操作为关闭 BFD 认证功能。

2. BFD 功能与 RIP（ng）协议联动配置

命令	解释
接口配置模式	
rip bfd enable no rip bfd enable	在指定接口上配置 RIP 协议与 BFD 联动, no 命令操作为禁用已经配置的 RIP 协议与 BFD 进行联动功能。
ipv6 rip bfd enable no ipv6 rip bfd enable	在指定接口上配置 RIPng 协议与 BFD 联动; no 命令操作为禁用已经配置的 RIPng 协议与 BFD 进行联动功能。

3. BFD 功能与静态路由（IPv6）协议联动配置

命令	解释
全局配置模式	
ip route {vrf <name> <ipv4-address> <ipv4-address>} mask <nextHop> bfd no ip route {vrf <name> <ipv4-address> 	配置静态路由与 BFD 进行联动。no 命令操作为取消配置静态路由与 BFD 进行联动。

<code><ipv4-address>} mask <nexthop> bfd</code>	
<code>ipv6 route {vrf <name> <ipv6-address> <ipv6-address>} prefix <nexthop> bfd</code> <code>no ipv6 route {vrf <name> <ipv6-address> <ipv6-address>} prefix <nexthop> bfd</code>	配置 IPv6 静态路由与 BFD 进行联动。 <code>no</code> 命令操作为取消配置 IPv6 静态路由与 BFD 进行联动。

4. BFD 功能与 VRRP（v3）协议联动配置

命令	解释
VRRP（v3）组配置模式	
<code>bfd enable</code> <code>no bfd enable</code>	启用 VRRP（v3）协议与 BFD 联动功能，在该组上启用 BFD 检测功能， <code>no</code> 命令操作为禁用 VRRP（v3）协议与 BFD 联动功能。

12.3 BFD举例

12.3.1 BFD与静态路由联动举例

案例：

在Switch A 上配置静态路由可以到达14.1.1.0/24网段路由，在Switch B上配置静态路由可以到达15.1.1.0/24网段路由，并都使能BFD检测功能；当Switch A 和Switch B 链路出现故障时BFD 能够快速感知。



配置步骤如下：

配置 switch A:

```
Switch#config
Switch(config)#interface vlan 12
Switch(config-if-vlan12)#ip address 12.1.1.1 255.255.255.0
Switch(config)#interface vlan 15
Switch(config-if-vlan15)#ip address 15.1.1.1 255.255.255.0
Switch(config)#ip route 14.1.1.0 255.255.255.0 12.1.1.2 bfd
```

配置 switch B:

```
Switch#config
```

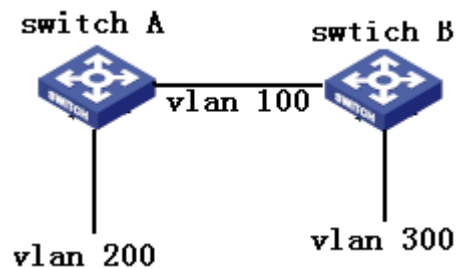
```
Switch(config)#interface vlan 12
Switch(config-if-vlan12)#ip address 12.1.1.2 255.255.255.0
Switch(config)#interface vlan 14
Switch(config-if-vlan15)#ip address 14.1.1.1 255.255.255.0
Switch(config)#ip route 15.1.1.0 255.255.255.0 12.1.1.1 bfd
```

Switch B 和二层交换机之间链路发生故障时，Switch A能够快速感知Switch B的变化。此时查看静态路由，路由处于Inactive 状态。

12.3.2 BFD与RIP路由联动举例

案例：

Switch A、Switch B连接并运行RIP协议，在两台交换机上开启BFD功能。当Switch A和Switch B链路出现故障时BFD能够快速感知。



配置步骤如下：

配置 switchA:

```
Switch#config
Switch(config)#bfd mode active
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#ip address 10.1.1.1 255.255.255.0
Switch(config)#interface vlan 200
Switch(config-if-vlan200)#ip address 20.1.1.1 255.255.255.0
Switch(config)#router rip
Switch (config-router)#network vlan 100
Switch (config-router)#network vlan 200
Switch(config)#interface vlan 100
Switch(config-if-vlan100) #rip bfd enable
```

配置 switchB:

```
Switch#config
Switch(config)#bfd mode passive
Switch(config)#interface vlan 100
Switch(config-if-vlan100)#ip address 10.1.1.2 255.255.255.0
Switch(config)#interface vlan 300
Switch(config-if-vlan300)#ip address 30.1.1.1 255.255.255.0
```

```
Switch(config)#router rip
```

```
Switch (config-router)#network vlan 100
```

```
Switch (config-router)#network vlan 300
```

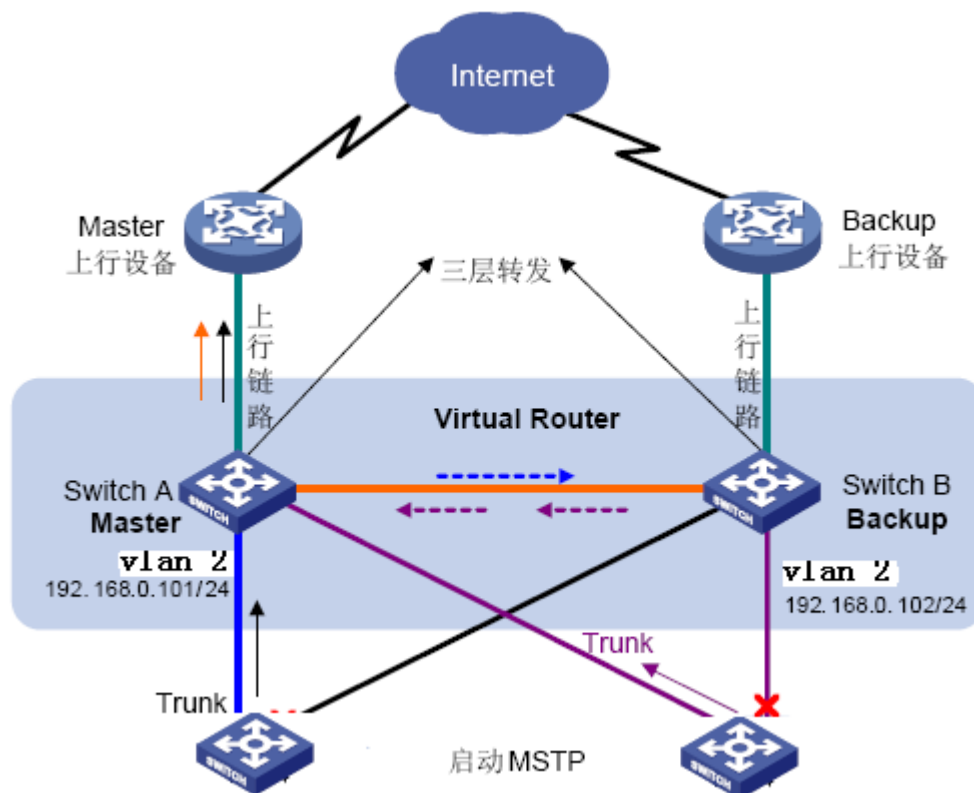
```
Switch(config)#interface vlan 100
```

```
Switch(config-if-vlan100) #rip bfd enable
```

Switch A 和 Switch B 交换机之间链路发生故障时，BFD 可以快速感知链路故障。BFD 协议通知 RIP 协议删除学习到的路由。

12.3.3 BFD与VRRP联动举例

案例：当Master出现故障时，只能依赖于Backup设置的超时时间来判断是否应该抢占，切换时间一般在3秒~4秒之间，无法达到秒级以下的切换速度。VRRP协议必须依赖一种可靠的链路检测机制来探测Master 的当前状态。Backup上的探测协议快速检测Master的运行状态，当Master状态出现故障时，Backup能够立即抢占成为Master，切换时间在100ms 以内。



配置修改：

配置步骤如下：

配置三层交换机Switch A

```
Switch#config
```

```
Switch(config)#bfd mode active
```

```
Switch(config)#interface vlan 2
```

```
Switch(config-if-vlan2)#ip address 192.16.0.101 255.255.255.0
```

```
Switch(config)#router vrrp 1
```

```
Switch(config-router)#virtual-ip 192.168.0.10
```

```
Switch(config-router)#interface vlan 1
```

```
Switch(config-router)#enable
```

```
Switch(config-router)#bfd enable
```

配置三层交换机Switch B

```
Switch#config
```

```
Switch(config)#bfd mode passive
```

```
Switch(config)#interface vlan 2
```

```
Switch(config-ip-vlan2)#ip address 192.16.0.102 255.255.255.0
```

```
Switch(config)#router vrrp 1
```

```
Switch(config-router)#virtual-ip 192.168.0.10
```

```
Switch(config-router)#interface vlan 1
```

```
Switch(config-router)#enable
```

```
Switch(config-router)#bfd enable
```

12.4 BFD排错帮助

当配置 BFD 功能出现问题时，请检查是否是如下原因：

- ☞ 路由协议的邻居是否已经成功建立，如果路由协议的邻居没有建立成功此时BFD无法进行检测
- ☞ 与静态路由联动时配置的source-ip是否正确，如果两端配置的IP不能互通BFD无法进行检测
- ☞ 与VRRP协议联动时，VRRP组是否已经成功建立，如果VRRP组没有建立成功此时BFD无法进行检测

第13章 BGP GR配置

13.1 GR简介

随着网络的发展，对于网络的可靠性也有了越来越高的要求，高可靠性技术（HA，High Availability）随之提出，即如何在路由器控制层面出现故障时仍然保证数据包能够正常转发，不影响网络的流量业务。

通常情况下，路由器故障后，其路由协议层面的邻居会检测到它们之间的邻居关系Down掉，然后过段时间再次Up，这个过程被称之为邻居关系震荡。这种邻居关系的震荡将最终导致路由震荡的出现，使得重启路由器在一段时间内出现路由黑洞或者导致邻居将数据业务从重启路由器处旁路，从而导致网络的可靠性大大降低。

为了实现交换机的高可靠性，需要上层的路由协议支持GR(Graceful Restart,优雅重启)的功能。使用路由协议的GR功能，可以在路由器的控制平面出现故障时，保证数据平面正常的包处理与转发。

GR功能的运用还能够减少路由震荡，减少控制平面资源消耗，提高网络稳定性。本文档所述的主要内容BGP路由协议的优雅重启（GR），即使BGP协议的重启过程尽可能的不影响转发过程，使数据包能够以正确的路径转发出去。

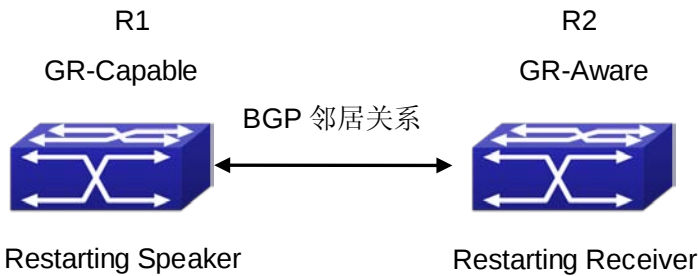


图 13-1 GR 的应用环境

BGP 路由协议的 GR 需要 GR-Capable 路由器与 GR-Aware 路由器协作完成，重启的路由器可以称为 Restarting Speaker（也可以叫做 GR-Restarter），而它的邻居都可以称为 Receiving Speaker（也可以叫做 GR-Helper）。Restarting Speaker 为 GR-Capable 路由器，Receiving Speaker 为 GR-Aware 路由器，这样才能协作完成 BGP 优雅重启的动作。假设路由器 R1、R2 建立 BGP 邻居关系，如图 13-1 所示，GR 的过程可以描述为：

对于 Restarting Speaker（GR-Restarter）来说：

- 1、在初始建立 BGP 邻居时，R1、R2 通过 OPEN 消息协商 GR 能力。
- 2、R1 重启发生，路由在接口板被保留，继续指导转发。
- 3、R1 同邻居 R2 重新建立 TCP 连接，在 BGP OPEN 消息中将 Restart state 置为 1，表明该路由器刚刚发生了重启。同时，通知邻居 restart time 的值（应该小于 OPEN 消息中的 Holdtime 时间）。另外，还需要告诉邻居，重启路由器支持哪一类地址族的路由的 GR。
- 4、R1 同邻居 R2 重新成功建立 BGP 邻居关系后，接收邻居发来的路由更新信息并处理，启动延期优选定时器。
- 5、R1 推迟本地的 BGP 路由计算过程，直到收到所有 GR-Aware 的 BGP 邻居的路由更新结束标志 End-of-RIB 或者等到本地的延期优选定时器超时。
- 6、执行路由计算，向各邻居发送路由更新；发送路由更新完毕后，向邻居发送 End-of-RIB 标志。

对于 Receiving Speaker（GR-Helper）来说：

- 1、R2与R1初始建立BGP邻居关系的过程中，与重启路由器协商GR能力，记录路由器R1为支持GR-Capable的路由器。
- 2、R1发生重启后，R2可能感知到和R1之间的TCP断掉，也可能在两者重新建立TCP连接前检测不到。如果没有检测到，执行4；否则，检测到执行3。
- 3、将重启路由器R1发过来的路由保持住，并且打上stale老化标志。启动Restart Timer。
- 4、中断旧的TCP连接，继续处理新的TCP连接，将重启路由器R1发过来的路由保持住，并且打上stale老化标志。启动Restart Timer。
- 5、与重启路由器建立新的邻居关系，删除Restart Timer，启动Stale Path Timer。
- 6、如果在建立新的邻居关系前，Restart Timer超时，或者收到新连接的OPEN消息中的 Restart flag不为1，或者OPEN消息中不含有对应的AFI/SAFI地址族支持信息，则清除被保持的重启路由器的路由。
- 7、向重启路由器发送路由信息更新，更新完毕，发送End-Of-RIB标志。
- 8、如果Stale Path Timer超时，清除被保持的重启路由器的路由。

13.2 GR配置任务序列

- 1. 配置是否支持 GR 能力
- 2. 配置特定邻居是否支持 GR 能力
- 3. 配置重启超时时间
- 4. 配置邻居的重启超时时间
- 5. 配置 BGP GR 保留废弃（STALE）路由超时时间
- 6. 配置 BGP GR 功能的优选延期超时时间

1. 配置是否支持 GR 能力

命令	解释
BGP 路由配置模式	

bgp graceful-restart no bgp graceful-restart	使能 BGP 支持 GR。
---	---------------

2. 配置特定邻居是否支持 GR 能力

命令	解释
BGP 协议单播地址族模式、VRF 地址族模式	
neighbor (A.B.C.D X:X::X:X WORD) capability graceful-restart no neighbor (A.B.C.D X:X::X:X WORD) capability graceful-restart	为邻居设置标记，在发送 OPEN 消息的时候携带 GR 的能力参数。

3. 配置重启超时时间

命令	解释
BGP 路由配置模式	
bgp graceful-restart restart-time <1-3600> no bgp graceful-restart restart-time <1-3600>	如果同时在 BGP 路由模式下和 BGP 邻居模式下配置该命令，以邻居模式下配置的命令为准。配置 BGP GR 的重启超时时间（Receiving Speaker 为发生重启的邻居启动一个重启超时定时器，使用 restart-time 为超时时间）。重启超时时间指定 Receiving Speaker 从发现邻居重启到收到邻居的 OPEN 消息最长的等待时间，如果超过这个时间 Receiving Speaker 没有收到邻居的 OPEN 消息，Receiving Speaker 可以删除为邻居保留的 SATLE 路由。

4. 配置邻居的重启超时时间

命令	解释
BGP 协议单播地址族模式、VRF 地址族模式	
neighbor (A.B.C.D X:X::X:X WORD) restart-time <1-3600> no neighbor (A.B.C.D X:X::X:X WORD) restart-time <1-3600>	配置邻居的重启超时时间，本命令 no 操作使该定时器恢复默认时间。

5. 配置 BGP GR 保留废弃（STALE）路由超时时间

命令	解释
BGP 路由配置模式	
bgp graceful-restart stale-path-time <1-3600> no bgp graceful-restart stale-path-time <1-3600>	BGP GR 功能的 stalepath-time 采用默认值 360s。stalepath-time 的默认值要比重启超时时间和优选延期超时时间要长的多，因为 Receiving Speaker 在从收到重启邻居的 OPEN 消息到收到该邻居的 EOR 这个时间段内不仅要完成向重启邻居发送初始路由更新的动作，还等待接收重启邻居的初始路由更新，直到接收完毕。

6. 配置 BGP GR 功能的优选延期超时时间

命令	解释
BGP 路由配置模式	
bgp selection-deferral-time <1-3600> no bgp selection-deferral-time <1-3600>	指定了 Restarting Speaker 从收到邻居的 OPEN 消息到收到所有邻居的 EOR 后开始路由优选计算的最长的等待时间，如果超过这个时间 Restarting Speaker 还没有收到所有邻居的 EOR，Restarting Speaker 可以开始路由的优选计算。

13.3 GR典型案例



图 13-2 GR 的应用环境

如图 13-2 所示，R1 与 R2 建立 bgp 邻居关系。R1 与 R2 断开邻居关系，R2 上的 BGP 协议会进入 helper 模式，保留 R1 传递给 R2 的路由表项，同时启动重启超时定时器，在定时器时间内收到 R1 的 open 消息或者定时器超时，在 R2 上被标记为 stale 的路由被删除。当 R1 重新与 R2 建立邻居关系后，R1 启动优选定时器，等待 R2 发送 EOR 消息或者定时器超时，R1 优选路由。R2 在收到 R1 的 open 消息后，启动废弃路由定时器，在接受 R1

发送的 EOR 或定时器超时，R2 删除定时器并删除 stale 路由。

R1 配置 int vlan 12, ip address 12.1.1.1

R2 配置 int vlan 12, ip address 12.1.1.2

R1 配置如下

```
R1#config
```

```
R1(config)#vlan 12
```

```
R1(config-vlan12)#int vlan 12
```

```
R1(config-if-vlan12)#ip address 12.1.1.1 255.255.255.0
```

```
R1(config-if-vlan12)#exit
```

```
R1(config)#router bgp 1
```

```
R1(config-router)#neighbor 12.1.1.2 remote-as 2
```

```
R1(config-router)#neighbor 12.1.1.2 capability graceful-restart
```

```
R1(config-router)#bgp selection-deferral-time 120
```

```
R1(config-router)#bgp graceful-restart restart-time 60
```

```
R1(config-router)#bgp graceful-restart stale-path-time 180
```

```
R1(config-router)#exit
```

R2 配置如下

```
R2#config
```

```
R2(config)#vlan 12
```

```
R2(config-vlan12)#int vlan 12
```

```
R2(config-if-vlan12)#ip address 12.1.1.2 255.255.255.0
```

```
R2(config-if-vlan12)#exit
```

```
R2(config)#router bgp 2
```

```
R2(config-router)#neighbor 12.1.1.1 remote-as 1
```

```
R2(config-router)#neighbor 12.1.1.1 capability graceful-restart
```

```
R2(config-router)#bgp selection-deferral-time 120
```

```
R2(config-router)#bgp graceful-restart restart-time 60
```

```
R2(config-router)#bgp graceful-restart stale-path-time 180
```

```
R2(config-router)#exit
```

第14章 OSPF GR配置

14.1 OSPF GR介绍

OSPF Graceful-Restart（简称 OSPF GR），即 OSPF 平滑重启，主要实现的功能是在路由协议重启或三层交换机发生主备切换时维持原有的数据转发正常，保证关键业务流量不中断，属于高可靠性（HA，High Availability）技术的一种。

目前，高端的三层交换机普遍采用了控制和转发分离的设计。负责路由协议计算的控制模块一般位于主控板上，而数据的转发则位于线卡，这样在主控板卡重启时，才有可能不影响线卡上的数据转发。因此，支持 GR 功能的设备，一般都是机架式设备，具有双主控板结构。

标准的 OSPF 协议（RFC2328）本身不支持 GR 技术，因此在由于各种原因出现主备切换或协议重启时，会造成网络流量中断和路由振荡。例如，在下图所示的拓扑环境中，S1 发生了主备倒换，S2 与 S1 之间的邻接关系将会丢失，这时 S2 会向 S3 和 S4 发送 Router-LSA，该 LSA 中将不包含 S1 与 S2 之间的链路。S3 和 S4 收到该 LSA 后会重新进行路由计算，计算出的转发路径不会包含 S1 与 S2 之间的链路。当 S1 完成主备倒换后，会重新与 S2 建立邻接关系，同步数据库，并导致 S2、S3、S4 需要重新进行路由计算。可见，当 S1 发生主备倒换会造成路由抖动，这对于一些性能要求较高的网络是不可接受的。

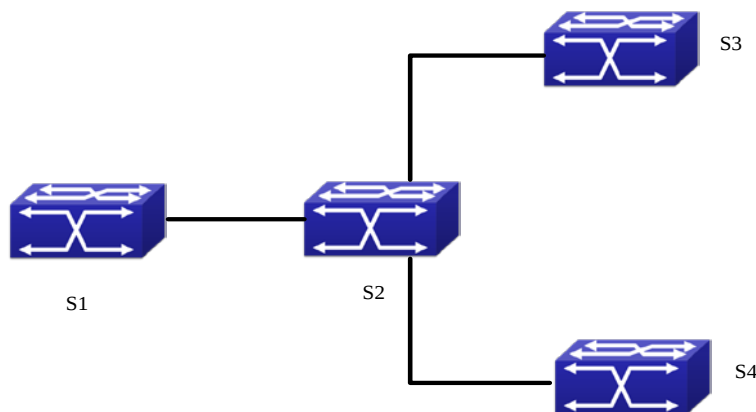


图 14-1 典型应用场景

RFC3623 中所描述 OSPF GR 就是针对以上情况提出的。它的基本思想是：如果在三层交换机主备倒换期间网络拓扑一直保持稳定，并且在这期间发生主备倒换的三层交换机可以保持它的转发表不变，那么该交换机的邻居可以维持与之的邻接关系不变，使得该交换机依然在转发路径上。因此如果 S1 与 S2 支持并启用了 GR 功能，那么在 S1 发生主备切换期间，S1 线卡会保持流量转发，S2 也将保持宣告与 S1 的邻居关系不变，同时 S3 及 S4 不会发现网络拓扑的变化，也不需要重新计算路由，这样就保证了 S1 主备切换期间的流量

转发，避免路由振荡。

按照三层交换机在 GR 过程中的不同作用可以将其分为 GR restarter 和 GR helper。GR restarter 是发生主备切换或进行协议重启的三层交换机，而 GR helper 是协助 GR restarter 进行 GR 的三层交换机。上例中 S1 是 GR restarter，S2 是 GR helper。

基于 OSPF GR 以上特征，OSPF GR 主要有以下优点：

- ☞ 提高网络可靠性
- ☞ 减少路由振荡对网络的影响
- ☞ 减少对业务流量的影响，避免切换期间的丢包

14.2 OSPF GR基本配置

OSPF GR 配置任务序列如下：

1. 使能 OSPF 进程的 GR 功能
2. 配置 OSPF GR restarter 的 grace-period（可选）
3. 配置 OSPF GR helper 的策略（可选）

1. 使能 OSPF 进程的 GR 功能

命令	解释
OSPF 协议配置模式	
capability restart graceful no capability restart	开启特定 OSPF 进程的 GR 功能(默认开启)。

2. 配置 OSPF GR restarter 的 grace-period（可选）

命令	解释
全局配置模式	
ospf graceful-restart grace-period <integer> no ospf restart grace-period	配置 GR restarter（进行主备切换或协议重启的交换机）的 grace period。本命令的 no 操作为恢复 restarter 的 grace period 为缺省值。

3. 配置 OSPF GR helper 的策略（可选）

命令	解释
全局配置模式	
ospf graceful-restart helper max-grace-period <integer> no ospf graceful-restart helper	GR helper 策略之一，配置 helper 支持的最大 grace period。本命令的 no 操作为删除所有配置的 helper 策略。
ospf graceful-restart helper never no ospf graceful-restart helper	GR helper 策略之一，配置交换机不能作为 OSPF GR helper 角色。本命令的 no 操作为删除所有配置的 helper 策略。

14.3 OSPF GR举例

案例：

四台交换机 S1-S4（都是双主控结构，支持 OSPF GR 功能），均使能 OSPF，实现以下功能：

- 1. 上游交换机 S1 在主备切换时保持流量转发，下游 S2-S4 不会出现路由振荡，网络流量不断；
- 2. S1 需要在 120S 内完成主备切换及协议重启，否则 S2 将退出 GR，重新计算路由；
- 3. S1 不充当 OSPF GR Helper 角色（即当 S2 发生主备切换或 OSPF 协议重启时，S1 不会协助其进行 GR，而是重新计算路由）

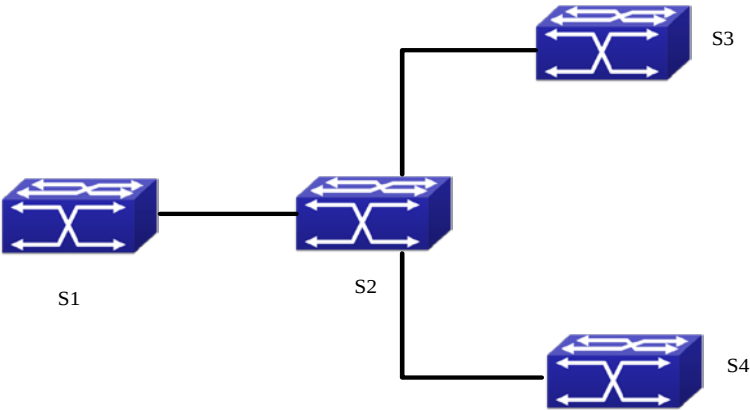


图 14-2 典型应用

配置步骤：交换机缺省已经使能 OSPF GR 功能，因此只需要配置 OSPF GR 的参数及 helper 策略即可。（下面只列出了与 OSPF GR 相关的配置，省略拓扑中 OSPF 的具体配置）

```
S1
S1(config)#ospf graceful-restart grace-period 120
S1(config)# ospf graceful-restart helper never
```

S2

```
S2(config)# ospf graceful-restart helper max-grace-period 120
```

14.4 OSPF GR排错帮助

当在使用 OSPF GR 过程中遇到问题，请检查是否是如下原因：

- GR restarter 交换机是否支持 OSPF GR 功能且是双主控结构，请确保没有关闭特定 OSPF 进程的 GR 功能
- 在 OSPF GR 过程中，是否出现网络拓扑变化，如果出现变化交换机有可能退出 GR，进行正常的 OSPF 重启
- 请确保 GR restarter 的所有邻居均支持 GR 功能
- 请尽量不要在交换机进行 GR 期间修改 OSPF 的相关配置