路由协议配置

目 录

第 1 章　IP 路由协议概述

在本公司的交换机中实现了多个 IP 路由动态协议，它们将在本文中各个协议的说明中分别予以介绍。

IP 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）。本公司交换机支持 RIP、OSPF、BGP 和 BEIGRP。您可以根据您的需要分别配置 RIP、OSPF、BGP 和 BEIGRP。在我们的交换机上，支持同时配置多个路由协议的进程，包括任意多个的 OSPF 进程（如果内存能够分配），一个 BGP 进程，任意多个 RIP 进程和任意多个的 BEIGRP 进程。您可以使用 redistribute 将其它路由协议的路由重新发布到当前的路由进程的数据库中，以此来将多个协议进程的路由联系起来。

为了配置 IP 动态路由协议，首先必须配置创建相应的进程，并且将相应的网络端口与一定的动态路由进程相联系起来，指定路由进程在哪些端口上启动。为此，你可能需要在相应的配置命令文档中查看相关的配置步骤。

本文中的路由设备，隐含指的是交换机。

1.1　选择路由协议

选择路由协议，这是一个复杂的过程。在选择路由协议的时候，你必须考虑如下的因素：

- 网络的大小以及复杂度
- 是否需要支持可变长网络
- 网络流量
- 安全的要求
- 可靠性的要求
- 策略
- 其它

在这里，我们并不深入的介绍这个问题，只是提醒用户注意，您所选择的路由协议必须要能够满足您网络的情况，适应您的需求。

1.1.1　内部网关路由协议

内部网关路由协议是用来在一个自治系统中的网络目标。所有的 IP 内部网关路由协议在启动的同时必须将其与一定的网络相联系起来（比如配置 network）。每个路由进程都监听网络上的来自其它交换机的更新报文，同时在网络上广播它自己的路由信息。本公司交换机支持的内部网关路由协议有：

- RIP
- OSPF
- BEIGRP

1.1.2 外部网关路由协议

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。一般要求配置相应的用来交换路由的邻居、公布为可到达的网络以及本地的自治系统号。本公司交换机支持的外部网关路由协议有 BGP。

第 2 章　VRF 配置

2.1　概述

VPN 的关键之一是保持安全，分隔数据；它必须防止不属于同一 VPN 的站点之间的通信。为了让 PE 设备上能区分是哪个本地接口上送来的 VPN 用户路由，在 PE 设备上创建了大量的虚拟路由设备，每个虚拟路由设备都有各自的路由表和转发表，这些路由表和转发表统称为 VRF(VPN Routing and Forwarding instances)。一个 VRF 包含了同一个站点相关的路由表、转发表、接口（子接口）、路由实例以及路由策略等。在 PE 设备上，属于同一个 VPN 的物理端口或逻辑端口对应一个 VRF。

2.2　VRF配置任务列表

如果想配置 VRF，需要完成下面的任务。

- 创建 VRF 表
- 将接口与 VRF 相关联
- 配置 VRF 的目标 VPN 扩展属性
- 配置 VRF 的描述说明
- 配置 VRF 静态路由
- 监视 VRF
- 维护 VRF
- VRF 配置举例

2.3　VRF配置任务

2.3.1　创建 VRF 表

要创建 VPN 路由和转发表，在全局配置模式下按以下步骤进行：

命令	目的
PE_config**#ip vrf** ce	进入 VRF 配置模式，定义 VRF 表。
PE_config_vrf_ce**#rd** *ASN:nn or IP-address:nn*	指定 VRF 的路由标记，创建 VRF 路由和转发表

PE_config_vrf_ce#**route-target** [**export** | **import** | **both**] *ASN:nn or IP-address:nn*	创建 VRF 的输入和输出目标 VPN 扩展属性

2.3.2 将接口与 VRF 相关联

将接口与 VRF 相关联，按以下步骤进行：

命令	目的
PE_config#**interface vlan** 1	进入接口配置模式。
PE_config_v1#**ip vrf forwarding** *vrf-name*	将接口与 VRF 相关联。
PE_config_v1#**ip address** *ip-address subnet-mask*	配置接口 IP 地址。

2.3.3 配置 VRF 的目标 VPN 扩展属性

配置 VRF 的目标 VPN 扩展属性，按以下步骤进行：

命令	目的
PE_config#**ip vrf** ce	进入 VRF 配置模式。
PE_config_vrf_ce#**rd** *ASN:nn or IP-address:nn*	配置 VRF 路由标记，创建 VRF 表。
PE_config_vrf_ce#**route-target** [**export** | **import** | **both**]*ASN:nn or IP-address:nn*	配置 VRF 的输入和输出目标 VPN 扩展属性。
PE_config_vrf_ce#**import map** *WORD*	配置加入 VRF 路由表中的路由的 route-map 过滤。
PE_config_vrf_ce#**export map** *WORD*	将符合 route-map 条件的目标 VPN 扩展属性加入到 VRF 的输出目标 VPN 扩展属性中。

在将本地路由发布给其它 PE 路由设备之前，入口 PE 将为从直连站点学习到的每条路由附加一个路由目标属性。附加到路由上的目标值是基于在输出目标扩展属性中配置的 VRF 的值。

在将由其它 PE 发布的远端路由安装在本地 VRF 前，出口 PE 路由设备上的每个 VRF 都将配置一个输入目标扩展属性。PE 路由设备只有在 VPN-IPv4 路由中承载的路由目标属性与某一 PE 路由设备上的 VRF 输入目标相匹配时，才会将该路由安装到某一 VRF 中。

2.3.4 配置 VRF 的描述说明

配置 VRF 的描述说明，按以下步骤进行：

命令	目的
PE_config#**ip vrf** ce	进入 VRF 配置模式。
PE_config_vrf_ce#**rd** *ASN:nn or IP-address:nn*	配置 VRF 路由标记，创建 VRF 表。
PE_config_vrf_ce# **description** LINE	配置 VRF 的描述说明。

2.3.5 配置 VRF 静态路由

配置 VRF 静态路由，按以下步骤进行：

命令	目的
PE_config#**ip vrf** ce	进入 VRF 配置模式。
PE_config_vrf_ce#**rd** *ASN:nn or IP-address:nn*	配置 VRF 路由标记，创建 VRF 表。
PE_config_vrf_ce#**exit**	退出 VRF 配置模式。
PE_config#**ip route** [**vrf** vrf-name] dest mask { type num | nexthop } [distance]	配置 VRF 静态路由。

2.3.6 监视 VRF

监视 VRF，可以显示 VRF 的统计信息。要监视，按以下步骤进行：

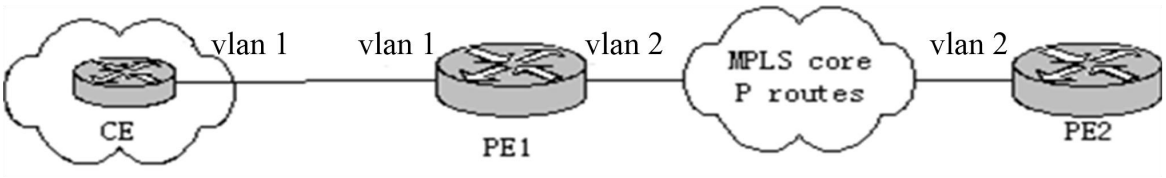
命令	目的
PE#**show ip vrf**	显示 VRF 及所关联的端口信息。
PE#**show ip vrf** [{**brief** | **detail** | **interfaces**}] *vrf-name*	显示 VRF 配置信息及相关联的端口信息。
PE#**show ip route vrf** *vrf-name*[A.B.C.D | all | beigrp | bgp | ospf | rip | connect | static | summary]	显示 VRF 路由表中路由信息。

2.3.7 维护 VRF

维护 VRF，跟踪主路由表和 VRF 路由表路由的变化及 VRF 配置信息的变化，在管理态下，按以下步骤进行：

命令	目的
PE**#debug ip routing**	跟踪主路由表中路由的添加、删除、改变等路由的变化情况。
PE **#debug ip routing message**	跟踪 VRF 接收和发送的信息。
PE **#debug ip routing vrf** *vrf-name*	跟踪指定 VRF 路由表中路由的添加、删除、改变等路由的变化情况。

2.4 VRF配置举例



路由设备的配置如下：

路由设备 CE：

```
interface loopback 0
   ip address 22.1.1.1 255.255.255.0
!
interface vlan 1
   ip address 170.168.20.152 255.255.255.0
!
router ospf 1
   network 170.168.20.0 255.255.255.0 area 0
   network 22.1.1.0 255.255.255.0 area 0
!
```

路由设备 PE1：

```
ip vrf pe1
        rd 1:1
        route-target 1:1
!
interface vlan 1
   ip vrf forwarding pe1
   ip address 170.168.20.153 255.255.255.0
!
interface vlan 2
   ip address 176.168.20.152 255.255.255.0
!
router ospf 1 vrf pe1
   network 170.168.20.0 255.255.255.0 area 0
!
router bgp 1
        neighbor 176.168.20.154 remote-as 2


        address-family vpnv4
        neighbor 176.168.20.154 activate
```

exit-address-family

address-family ipv4 vrf pe1

no synchronization

redistribute ospf 1

exit-address-family

路由设备 PE2:

```
ip vrf pe2
    rd 1:1
    route-target 1:1
!
interface loopback 0
   ip vrf forwarding pe2
   ip address 44.1.1.1 255.255.255.0
!
interface vlan 2
   ip address 176.168.20.154 255.255.255.0
!
router bgp 2
    neighbor 176.168.20.153 remote-as 1
```

address-family vpnv4

neighbor 176.168.20.153 activate

exit-address-family

address-family ipv4 vrf pe2

no synchronization

redistribute connected

exit-address-family

第 3 章　静态路由配置

3.1　概述

静态路由是一种特殊的路由，由管理员手工配置。在组网结构比较简单的网络中，只需配置静态路由就可以实现网络互通。恰当地设置和使用静态路由可以改善网络的性能，并可为重要的网络应用保证带宽。

静态路由的缺点在于：不能自动适应网络拓扑结构的变化，当网络发生故障或者拓扑发生变化后，可能会出现路由不可达，导致网络中断，此时必须由网络管理员手工修改静态路由的配置。

默认路由是在路由设备找不到匹配的路由表项时才使用的路由：如果报文的目的地址不能与路由表中的任何表项相匹配，那么该报文将选择默认路由；如果没有默认路由且报文的目的地不在路由表中，那么该报文将被丢弃。默认路由可以通过静态路由配置，以到网络 0.0.0.0/0 的形式在路由表中出现。

3.2　静态路由配置任务列表

如果想配置静态路由，需要完成下面的任务。

- 配置相关接口的物理参数
- 配置相关接口的链路层属性
- 配置相关接口的 IP 地址

3.3　静态路由配置任务

3.3.1　配置静态路由

要配置静态路由，需要进入全局配置模式，按以下步骤进行：

命令	目的
ip route *A.B.C.D mask* {next-hop | interface} [distance] [tag tag] [global] [description]	配置静态路由。

3.4　静态路由配置举例

指定到网段 10.0.0.0/8 的报文的出端口为 interface vlan 1，配置如下：

ip route 10.0.0.0 255.0.0.0 vlan 1

第 4 章　RIP 配置

4.1　概述

路由信息协议 RIP 是一个相对过时但仍然普遍使用的内部网关协议（IGP），主要应用于规模较小的同质型网络。RIP 是一个经典的距离向量路由协议，出现在 RFC 1058 中。

RIP 通过用户数据报协议 UDP 数据分组的广播来交换路由信息。在路由设备中，每 30 秒钟发送一次路由信息的更新。如果一台路由设备在 180 秒内没有接收到来自相邻路由设备的更新，便把路由表中来自该路由设备的路由标记为“不可用的”。如果接下去的 120 秒内仍然没有接收到更新，便把这些路由从路由表中删除。

RIP 使用跳跃计数（hop count）来作为衡量不同路由的权值。这个跳跃计数是指一个分组从信源到达信宿所经过的路由设备的数目。直接相连网络的路由权值为 0，不可到达网络的路由权值为 16。由于 RIP 使用的路由权值范围较小，所以对大规模的网络就显得不大适合。

如果路由设备有一条缺省路由，RIP 就宣告通向伪网络 0.0.0.0 的路由。实际上，网络 0.0.0.0 并不存在，它只是用来在 RIP 中实现缺省路由的功能。如果 RIP 学习到了一条缺省路由，或者 RIP 中设置了默认网关，路由设备都将宣告这个缺省的网络。

RIP 向指定的接口发送路由更新。如果一个接口没有被指定，那么 RIP 不会向该接口宣告路由更新。

本公司路由设备的 RIP-2 支持明文和 MD5 及动态认证、路由汇总、无分类域间路由（CIDR）和变长子网掩码（VLSM）。

4.2　RIP配置任务列表

如果想配置 RIP，必须完成下面的任务。你首先要建立 RIP 实例，其余的任务则是可选的。

- 启动 RIP 实例
- 生成 RIP 实例接口
- 允许 RIP 路由更新分组的单目广播
- 对路由权值应用偏移量
- 调整计时器
- 指定 RIP 版本号
- 激活 RIP 认证

- 激活接口的“聋”“哑”状态
- 禁止路由汇总
- 禁止信源 IP 地址认证和零域验证
- 最大等价路由数
- 激活或禁止水平分割
- 监视和维护 RIP

4.3 RIP配置任务

4.3.1 启动 RIP 实例

要激活 RIP 实例，进入全局配置模式，按以下步骤进行：

命令	目的
router rip *process-id* [**vrf** *vrf-name*]	建立RIP实例，进入路由设备RIP实例配置模式。

4.3.2 生成 RIP 实例接口

启动 RIP 实例后，只有接口关联到实例下才能生成 RIP 的直连网段，以及用这些接口和邻居交换路由信息。实例要关联接口，在接口的配置态，按以下步骤进行：

命令	目的
router rip *process-id* **enalbe**	将接口关联到process-id实例中。

若要使接口成为 RIP 的激活接口（生成接口的直连路由，接口能够收发 RIP 协议报文），需要满足：接口关联到 RIP 实例，接口上有合法的 ip 地址，接口的状态 up。

另外，当在一个接口中 enable 一个 RIP 实例的时候，若该接口被指定了 vrf，而实例的 vrf 和接口上的 vrf 不一致，该接口不能够成为 RIP 的激活接口，直到修订了接口的 vrf。

当一个接口关联了一个尚未创建的 RIP 实例，将会以接口上的 vrf（若被指定了 vrf）和 enable 的 process-id 创建 RIP 实例。

每个接口只能属于一个 RIP 实例。

4.3.3 允许 RIP 路由更新分组的单目广播

RIP 通常是一个广播型协议，如果要使 RIP 路由更新能够到达非广播型网络，你必须对路由设备进行配置以允许路由信息的交换。要达到这样的目的，在路由配置模式中使用如下命令：

命令	目的
neighbor *ip-address*	定义一个邻居路由设备，与之交互路由信息。

另外，如果你想控制哪些接口可以用来交换路由信息，可以使用命令 ip rip passive 指定在某个（某些）接口上禁止路由更新的发送。如果参考在“配置 IP 路由中与协议无关的命令”一章中的“过滤路由信息”中关于路由过滤的讨论。

4.3.4 对路由权值应用偏移量

偏移量列表是用来对那些由 RIP 学习到的入站和出站路由增加一个偏移量。这就提供了一个本地的机制来增加路由权值。另外，你还可以使用访问列表或接口来限制偏移量列表。要增加路由权值，在路由配置模式中使用如下命令：

命令	目的
offset-list ｛*interface-type number*	* ｝ **{in|out}** *access-list-name offset*	对路由权值增加一个偏移量。

4.3.5 调整计时器

路由协议使用几个计时器来判断发送路由更新的频率、多长时间内路由变为无效以及其它参数。你可以调整这些计时器使路由协议的性能更适合网络互联的需要。

也有可能调整路由协议来加速各种 IP 路由算法的收敛时间，做到向冗余路由设备的快速备份，以达到在需要快速恢复（quick recovery）的场合中向终端用户保证最小的崩溃时间。要调整计时器，在路由配置模式中使用如下命令：

命令	目的
timers holddown *value*	经过多长时间（单位：秒）从路由表中删除某一条路由。
timers expire *value*	经过多长间隔（单位：秒）路由被宣告为无效。
timers update *value*	发送路由更新的频率（发送更新的间隔，单位：秒）。
timers trigger *value*	触发更新的间隔（单位：秒）
timers peer *value*	peer超时间隔（单位：秒）

4.3.6 指定 RIP 版本号

本公司路由设备的 RIP-2 支持认证、密钥管理、路由汇总、无分类域间路由（CIDR）和变长子网掩码（VLSM）。

缺省情况下（没有配置全局和端口上的 version），路由设备接收 RIP-1 和 RIP-2 的分组，向外发送分组的时候，若端口有 peer 则依据 RIP 的自适应规则（一个端口上若只有一个 peer 或者有多个 peer 但 peer 彼此的 version 都一致，则按 peer 的 version 响应；若同一个端口上有多个peer但peer彼此的version不一致，则依据系统的RIP缺省情况处理），若没有 peer 则发送缺省的 RIP-2 的分组。通过配置，可以使路由设备仅发送和接收 RIP-1

的分组，或者仅发送和接收 RIP-2 的分组。要达到这样的目的，在路由配置模式中使用如下命令：

命令	目的
version {1 | 2}	配置路由设备仅仅发送和接收RIP-1或RIP-2的分组。

上述的任务控制 RIP 的缺省行为。你也可以配置某一特定接口来改变这个缺省行为。要控制接口发送 RIP-1 分组还是 RIP-2 分组，以及要控制发送 request 的时候的分组，在接口配置模式下使用如下命令：

命令	目的
ip rip send version 1	配置接口仅仅发送RIP-1的分组。
ip rip send version 2	配置接口仅仅发送RIP-2的分组。
ip rip send version compatibility	广播发送RIP-2更新报文。
ip rip v1demand	在发送request的时候发送RIP-1的分组。
ip rip v2demand	在发送request的时候发送RIP-2的分组。

同样，要控制接口接收 RIP-1 分组还是 RIP-2 分组，在接口配置模式下使用如下命令：

命令	目的
ip rip receive version 1	配置接口仅仅接收RIP-1的分组。
ip rip receive version 2	配置接口仅仅接收RIP-2的分组。
ip rip receive version 1 2	配置接口可以接收RIP-1和RIP-2的分组。

4.3.7 激活接口的“聋”“哑”状态

在缺省情况下 RIP 覆盖的接口是可以发送和接收路由更新的，通过配置可以灵活应用 RIP 协议，让 RIP 接口处于只接收不发送或者只发送不接收状态。

要配置接口的聋哑状态，要在接口配置模式下进行：

命令	目的
Ip rip passive	接口不发送rip协议分组。
ip rip deaf	接口不接收rip协议分组。

4.3.8 激活 RIP 认证

RIP-1 不支持认证。如果你在发送和接收 RIP-2 的分组，你可以在接口上激活 RIP 认证功能。

在 RIP 激活的接口上，我们支持多种认证模式：明文认证、MD5 认证、动态认证（md5 和 sha1）。每个 RIP-2 分组中缺省时使用明文认证。

注意：

如果处于安全的目的，不要在 RIP 分组中使用明文验证，这是因为未经加密的认证密钥发送在每个 RIP-2 分组中。如果在不涉及安全问题的情况下（例如：要保证配置错误的主机不能参与路由），可以使用明文验证。

要配置 RIP 明文认证，在接口配置模式下，按如下步骤进行：

命令	目的
ip rip authentication simple	配置接口使用明文认证。
ip rip password *string*	配置明文认证密钥。

要配置 RIP 的 MD5 认证，在接口配置模式下，按如下步骤进行：

命令	目的
ip rip authentication md5	配置接口使用MD5认证。
ip rip md5-key *key-ID* **md5** *key*	配置MD5认证密钥和认证ID 。

要配置 RIP 的动态认证，在接口配置模式下，按如下步骤进行：

命令	目的
ip rip authentication dynamic	配置接口使用动态认证（md5和sha1）。
ip rip dynamic-key *key-ID* { **md5 | sha1** } *key xxxx-xx-xx-xx:xx xx:xx*	配置动态认证密钥和认证ID。

RIP 认证配置完毕后，建议在接口配置模式下，按如下步骤进行：

命令	目的
ip rip authentication commit	认证通不过时尽快老化对端peer和从对端学到的路由。

4.3.9 禁止路由汇总

RIP-2 在缺省情况下支持自动路由汇总。在穿越有类别网络边界时汇总 RIP-2 路由。RIP-1 自动路由汇总功能总处于激活状态。

如果有一个分离的子网，就要禁止路由汇总来宣告这个子网。如果路由汇总被禁止，在穿越由类别网络边界时，路由设备将传送子网和主机的路由信息。在路由配置模式下，使用下面命令来禁止自动汇总。

命令	目的
no auto-summary	禁止自动汇总。

4.3.10 禁止可信源 IP 地址验证和零域验证

缺省情况下，路由设备会对收到的 RIP 路由更新的可信源 IP 地址进行验证。如果这个地址非法，将丢弃这个路由更新。

如果你有一台路由设备希望接收来自它的更新，但是你并没有在接收端的路由设备的接口上关联对应的 rip 实例，那么就可以禁止这个功能。但在正常情况下，推荐你不要使用这个命令。在路由配置模式下使用如下命令将禁止对入站路由更新的可信源 IP 地址进行验证的缺省功能。

在缺省情况下，路由设备会对收到的 version 1 下路由条目的必须为零的字段进行验证。如果相应的字段不能通过零域的合法性检验，该路由条目将被丢弃。如果配置不进行该项验证，将可能导致本地学习到对端发送过来的错误的路由信息

命令	目的
no validate-update-source	禁止对入站RIP路由更新的可信源IP地址的验证。
no check-zero-domain	禁止对入站RIP路由更新的必须为零字段的验证。

4.3.11 最大等价路由数

在缺省情况下，RIP 路由信息最多允许生成 4 条等价路由。当从多个邻居学习到某个或某些相同网段的路由信息生成等价路由时，若某个网段的等价路由数大于当前最大等价路由数将不能添加到 RIP 的 database 中。

在路由配置模式下使用如下命令对 RIP 本地路由表中最大等价路由数进行配置：

命令	目的
maximum-nexthop *number*	配置RIP路由信息最大等价路由数 。
No maximum-nexthop	恢复缺省最大等价路由数 。

4.3.12 激活或禁止水平分割

正常情况下，与广播型 IP 网络相连并使用距离向量路由协议的路由设备采用水平分割机制来减小路由环路的可能性。水平分割阻塞路由信息向接收到该路由信息的接口进行宣告。这样做可以优化多个路由设备间的通信（尤其是在环路打破的时候）。但是，对于非广播型网络（例如帧中继），情况并非那么理想。此时，你可能要禁止水平分割。

如果一个接口配置了辅助的 IP 地址并且激活了水平分割，路由更新的信源 IP 地址可能不包括每一个辅助地址。一条路由更新中的信源 IP 地址只包括一个网络号（除非水平分割被禁止）。

要激活或禁止水平分割，在接口配置模式下使用如下命令：

命令	目的
ip rip split-horizon {simple | poisoned}	激活水平分割。
no ip rip split-horizon {simple | poisoned}	禁止水平分割。

在缺省情况下，对于点对点接口，水平分割是激活的；对于点对多点接口，水平分割是禁止的。可选参数 simple 和 poisoned 分别表示简单水平分割和带毒性逆转的水平分割。

请参考本章中“水平分割举例”一节中关于使用水平分割的具体例子。

注意：

在一般情况下，推荐你不要改变缺省状态，除非你能肯定你的应用程序需要状态的改变才能正确地宣告路由。记住：如果水平分割在一个串行接口上被禁止（并且与该接口相连的是一个分组交换网），你必须对该网络上任何相关多目广播组中的路由设备禁止水平分割。

4.3.13 监视和维护 RIP

监视和维护 RIP，可以显示网络的统计信息，如：RIP 协议参数配置、使用网络、网络通信实时跟踪等。这些信息能帮助你判断网络资源的利用，解决网络问题。能了解网络节点的可达性。

在管理态输入命令，使用下面的命令，可以显示各种路由统计信息。

命令	目的
show ip rip	显示所有RIP实例当前状态。
Show ip rip *process-id*	显示具体某个RIP实例的当前状态
show ip rip *process-id* **database**	显示某个RIP实例的所有路由。
show ip rip *process-id* **protocol**	显示某个RIP实例协议相关信息。
Show ip rip *process-id* **interface**	显示某个RIP实例全部的接口及接口状态
show ip rip *process-id* **peer**	显示某个RIP实例全部的peer及其状态

在管理态输入命令，使用下面的命令，可以跟踪路由协议信息。

命令	目的
debug ip rip database	跟踪RIP路由加入路由表、从路由表中删除路由、路由改变等过程信息。
debug ip rip packet [send | receive]	跟踪RIP协议报文。
debug ip rip message	跟着RIP事件，如定时器超时

4.4 RIP配置举例

两台交换机 A 和 B，配置如下：

交换机 A：

```
interface vlan1
ip address 192.168.20.81 255.255.255.0
ip rip 1 enable
```

```
!
interface loopback 0
ip address 10.1.1.1 255.0.0.0
ip rip 1 enable
!
router rip 1
!
```

交换机 B：

```
interface vlan1
ip address 192.168.20.82 255.255.255.0
ip rip 1 enable
!
interface loopback 0
ip address 20.1.1.1 255.0.0.0
ip rip 1 enable
!
router rip 1
!
```

第 5 章　BEIGRP 动态路由协议配置

5.1　BEIGRP路由协议简介

BEIGRP 使用的技术类似于距离向量协议：

- 路由设备只用直接连接的邻居提供的信息作出路由决策。
- 路由设备只向直接连接的邻居提供它所使用的路由信息。

但是，BEIGRP 与距离向量有一些主要差别，使它具有更多优点：

- BEIGRP 保存拓扑表中所有邻居发来的所有路由，而不只是保存迄今为止收到的最佳路由。
- BEIGRP 在无法访问目的地而又没有替换路由时能对邻居进行查询，因此 BEIGRP 的收敛速度可以和最佳链路状态协议相媲美。

DUAL（Diffused Update Algorithm）即扩散更新算法的提出是 BEIGRP 优于其他传统的距离向量路由协议的关键所在。它总是非常积极的工作，在无法访问目的地而又没有替换路由（可行后续者）时能对邻居进行查询。由于汇总过程是主动的而不是被动的（被动的等待路由超时），因此 BEIGRP 的汇合速度很快。

BEIGRP 是为了适应 EIGRP 的要求而设计的专有传输协议，直接建立在 IP 之上。它满足了 BEIGRP 的如下要求：

- 通过 Hello 报文动态的发现新邻居及旧邻居的消失。
- 所有数据传输均可靠。
- 传输协议允许单目广播和多目广播数据传输。
- 传输协议本身可以适应网络条件变化和邻居响应性变化。
- BEIGRP 可以根据要求限制自己占用带宽的比率。

5.2　BEIGRP配置任务列表

要完成 BEIGRP 的配置需要完成下面的任务，其中激活 BEIGRP 协议是必须进行的，其他任务可以根据需要决定是否进行。

- 激活 BEIGRP 协议
- 配置占用带宽的百分比
- 调整 BEIGRP 复合距离的计算系数

- 应用 offset 调整路由的复合距离
- 配置转发路由
- 配置 BEIGRP 其他参数
- 监视与维护 BEIGRP 的运行

5.2.1 激活 BEIGRP 协议

要创建一个 BEIGRP 进程，需要依次执行下列命令：

命令	目的
router beigrp *as-number*	在全局配置态下增加一个BEIGRP进程。
network *network-number network-mask*	在路由配置态下增加网段到这个BEIGRP进程中。

完成上面的配置后，BEIGRP 将开始在属于此网段的所有接口上运行。通过 Hello 发现新邻居，通过 Update 进行初始路由交互。

5.2.2 配置可以占用的带宽百分比

在缺省状态下，BEIGRP 分组最多使用线路带宽的 50%。你可能希望改变这个缺省值，以保证其他数据的正常交互，或者接口使用 bandwidth 命令配置了与实际不符的端口带宽，希望通过命令来调整 BEIGRP 真正可使用的带宽。在这些情况下，你可以在接口配置态中使用下面的命令：

命令	目的
ip beigrp bandwidth-percent *percent*	配置BEIGRP报文使用的带宽占总带宽的最大百分比。

5.2.3 调整 BEIGRP 复合距离的计算系数

在某些情况下，可能需要调整 BEIGRP 复合距离的计算系数，从而最终影响路由的选路策略。虽然 BEIGRP 使用的缺省计算系数已经可以满足大多数网络环境，但在某些特殊条件下仍需要对其进行调整。但是这个调整可能会使整个网络产生巨大的变化，所以一定要由有丰富经验的工程师负责实施。

在路由配置态下使用下面的命令即可：

命令	目的
metric weights *k1 k2 k3 k4 k5*	调整BEIGRP复合距离计算系数。

5.2.4 应用 offset 调整路由的复合距离

我们使用偏移量列表可以根据需要有目的的增加所有入站和出站路由，或者其中符合要求的某几条路由的复合距离。这样做的目的是为了最终影响路由设备的选路结果，以达到我们期望的结果。在配置过程中，可以根据你的需要有选择性的在偏移量列表中指定访问列表或应用接口，以进一步明确需要进行增加偏移量操作的路由。请看下面的命令：

命令	目的
offset{*type number* | **} {**in** | **out**} *access-list-name offset*	应用一个偏移量列表 。

5.2.5 配置转发路由

BEIGRP 转发其他类型路由时，它遵循如下规则：

- 如果预转发的路由是静态路由或者直连路由，不需要配置“default-metric”命令，其复合距离相关参数（带宽、延迟、可靠性、有效负载和 MTU）直接从所在端口获得；
- 如果预转发的路由是 BEIGRP 协议的其他进程的路由，不需要配置“default-metric”命令，其复合距离相关参数直接从 BEIGRP 所在进程获得；
- 转发其他协议（如：rip，ospf）的路由，必须配置“default-metric”命令，其转发路由的符合距离由“default-metric”命令的配置值确定，若未曾配置“default-metric”命令则转发不起作用。

在同时运行 BEIGRP 协议和 RIP 协议的路由设备上，如果希望 BEIGRP 邻居能够学习到本路由设备中 RIP 协议学习的路由，那么就必须使用以下命令：

命令	目的
default-metric *bandwidth delay reliability loading mtu*	配置转发路由的缺省向量距离。
redistribute *protocol [process]* [**route-map** *name*]	进行转发路由至BEIGRP协议。

5.2.6 配置 BEIGRP 其他参数

为了适应不同的网络环境，使 BEIGRP 能够更有效，更充分的发挥作用，我们可能还需要对下面的一些参数进行调整：

- 调整 BEIGRP 发送 hello 报文的时间间隔以及邻居超时死亡时间
- 关闭水平分割

1. 调整 BEIGRP 发送 hello 报文的时间间隔以及邻居超时死亡时间。

BEIGRP hello 协议实现正确的 BEIGRP 操作所需的三个目标：

- 它发现能够访问的新邻居。邻居发现是自动的，不需要其他手工配置。

- 它验证邻居配置并只允许与以兼容方式配置的邻居通信。
- 它持续监视邻居可用性并探测邻居的消失。

路由设备在运行 BEIGRP 的所有接口上发送 hello 多目广播分组。每个支持 BEIGRP 的路由设备都接收这些多目广播分组，因此可以发现所有邻居。

Hello 协议用两个定时器探测邻居的消失：hello 间隔（hello interval）指定路由设备的接口上发送 BEIGRP hello 报文的频度，而保持定时器（hold timer）指定路由设备从指定邻居接收不到通信数据时等待多长时间再声明该邻居死亡。我们规定每次从相邻路由设备收到任意类型的 BEIGRP 分组时都要复位保持定时器。

不同的网络类型或网络带宽将使用不同的 hello 定时器缺省值：

接口类型包装		Hello 定时器（秒）	保持定时器（秒）
LAN接口	任意	5	15
WAN接口	HDLC或PPP	5	15
	NBMA接口，带宽<= T1	60	180
	NBMA接口， 带宽 > T1	5	15
	NBMA接口上的点对点子接口	5	15

Hello 协议中定时器缺省值的不同可能导致连接相同 IP 子网的 BEIGRP 邻居使用不同的 hello 和保持定时器。要解决这个问题，每个路由设备的 hello 分组中指定自己的保持定时器，每个 BEIGRP 路由设备用邻居的 hello 分组中指定的保持定时器判断这个邻居的超时。这样就可以使不同邻居故障探测定时器可以出现在同一个 WAN 云图的不同站点中。但是在一些特殊情况下，定时器的缺省值并不能满足，所以如果希望调整发送 hello 报文的时间间隔，使用下面的命令：

命令	目的
ip beigrp hello-interval *seconds*	调整发送此接口上发送hello报文的时间间隔。

如果希望调整邻居的超时定时器，使用下面的命令：

命令	目的
ip beigrp hold-time *seconds*	调整邻居的超时死亡时间。

2. 关闭水平分隔

一般情况下，我们希望使用水平分割。它将阻止从一个接口收到的路由信息再从同一个接口广播出去，从而避免路由循环。但在某些情况下，这样做并不是最优选择，那么我们可以使用下面的命令关闭水平分隔：

命令	目的
no ip beigrp split-horizon	关闭水平分隔。

5.2.7 监视和维护 BEIGRP

要清除与所有邻居的相邻性，使用下面的命令：

命令	目的
clear ip beigrp neighbors [*as-number* | *interface*]	清除邻居的相邻性。

使用下面的命令可以显示 BEIGRP 的各种统计信息：

命令	目的
show ip beigrp interface [*interface*] *[as-number*]	显示BEIGRP的接口信息。
show ip beigrp neighbors [*as-number* | *interface*]	显示BEIGRP的邻居信息。
show ip beigrp topology [*as-number* | **all-link | summary | active**]	显示BEIGRP的拓扑表信息。

5.3 BEIGRP配置举例

下面的实例配置了在 vlan 1 上发送 10.0.0.0/8 网段的汇总路由，属于此网段的所有子网路由将不会从此接口通知给邻居。同时，我们关闭了 BEIGRP 进程的自动汇总。

```
interface vlan 1
ip beigrp summary-address 1 10.0.0.0 255.0.0.0
!
router beigrp 1
network 172.16.0.0 255.255.0.0
no auto-summary
```

第 6 章　OSPF 配置

6.1　概述

OSPF 是 IETF 的 OSPF 工作组的开发的 IGP 路由协议。 为 IP 网络设计的 OSPF 支持 IP 子网和外部路由信息标记，也允许报文的认证以及支持 IP 多播。

本公司路由设备 OSPF 功能的实现遵守 OSPF V2 的要求（参见 RFC2328）。下面列出了实现中的一些关键特征：

- Stub 域——支持 stub 域。
- 路由转发——即被任何一种路由协议学习生成的路由都可以被转发到其他路由协议域。在自治域内，这表示 OSPF 能输入 RIP 学习到的路由。OSPF 学习到的路由也可以输出到 RIP。在自治域间，OSPF 能输入 BGP 学习到的路由； OSPF 路由也能输出到 BGP 中去。
- 认证——在一个域内的邻接路由设备之间，支持明文与 MD5 认证。
- 路由接口参数——可配置的接口参数有：输出花费、重传间隔、接口传输时延、路由设备的优先级别、判定路由设备的关机的时间间隔与 hello 包的时间间隔以及认证密钥。
- NSSA 区---参见 RFC 1587。
- 按需电路上的 OSPF---RFC 1793。

6.2　OSPF配置任务列表

OSPF 要求在全部域内路由设备、ABR 与 ASBR 之间进行交换路由数据。为了简化配置，可以让它们全部工作在默认参数，不需认证等；但如果要修改某些参数，则必须保证在所有路由设备上的参数一致。

为了能配置 OSPF， 完成下面的任务。除了激活 OSPF 是必须的外，其他配置都是可选配置。

- 启动 OSPF
- 配置 OSPF 的接口参数
- 配置 OSPF 网络类型
- 配置点到多点、广播网络
- 配置非广播类型网络

- 配置 OSPF 域参数
- 配置 OSPF 的 NSSA 域
- 配置 OSPF 域内路由的汇总
- 配置转发路由的汇总
- 生成默认路由
- 在 LOOPBACK 接口上选择路由 ID
- 配置 OSPF 的管理距离
- 配置路由计算的计时器
- 启动 On-Demand 链路的配置
- 监视和维护 OSPF

另外，配置路由转发， 可以参见配置独立协议的 IP 路由协议的特性配置的转发路由信息的有关内容。

6.3 OSPF配置任务

6.3.1 启动 OSPF

与其他的路由协议一样，激活 OSPF 要求创建 OSPF 路由进程，要求分配一个与处理过程相关的 IP 地址范围，分配一个与 IP 地址范围相关的区域 ID。在全局配置模式下，使用下面的命令：

命令	目的
router ospf *process-id*	这个命令激活OSPF路由协议，并且进入路由配置模式。
network *address mask* **area** *area-id*	这个命令配置OSPF 运行的接口以及接口的区域ID。

6.3.2 配置 OSPF 的接口参数

在 OSPF 实现中，允许按照需要修改接口有关的 OSPF 参数。并不需要改变任何一个参数，但必须保证某些参数在相连网络的所有路由设备上的保持一致。

在接口配置模式，使用下面的命令配置接口参数：

命令	目的
ip ospf authentication	配置OSPF 接口收发包的认证方式。
ip ospf cost *cost*	配置OSPF 接口发送包的权值。

ip ospf retransmit-interval *seconds*	属于同一个OSPF接口的邻居之间重传LSA的秒数。
ip ospf transmit-delay *seconds*	配置在一个OSPF接口传输LSA的估计时间（秒为单位）。
ip ospf priority *number*	配置路由设备成为OSPF DR路由设备的优先值。
ip ospf hello-interval *seconds*	配置在OSPF接口发送hello 包的时间间隔。
ip ospf dead-interval *seconds*	在这个规定的时间间隔内，未收到邻居的hello包，则认为邻居路由设备已关机。
ip ospf password *key*	为一个网段内的邻接路由的认证口令。它使用OSPF的简单的口令认证。
ip ospf message-digest-key *keyid* **md5** *key*	要求OSPF 使用MD5 认证。
ip ospf passive	在端口上不发送HELLO报文。

不同的物理网络上的 OSPF 配置，OSPF 把网络的物理媒体，分成以下三类：

- 广播网络(Ethernet, Token Ring, FDDI)
- 非广播、多访问网络 (SMDS, Frame Relay, X.25)
- 点对点网络 (HDLC, PPP)

能配置你的网络或者广播网络或者是非广播、多访问网络。

X.25 和帧中继网络提供了可选的广播能力，能通过 map 命令配置 OSPF 工作在广播网络。Map 命令可以参见广域网命令参考中有关 x.25 与帧中继的 map 命令的描述。

6.3.3 配置 OSPF 网络类型

不管网络的物理媒体类型，你都可以配置你的网络或者为广播网或者非广播、多访问网络。使用这个特性，你能灵活配置网络，可以将物理上的广播网络配置成非广播、多访问网络；也能配置非广播网络（X.25, Frame Relay, 与 SMDS）成为广播网络。这个特征也减少对邻居的配置，具体参见为非广播网络配置 OSPF 相关内容。

配置非广播、多访问网络为广播网络或者非广播网络，即假设从每一个路由设备到其他路由设备都存在虚链路，或假设为一个全网状网络。由于花费的限制，这常常是不现实的；或者有一个部分网状网。这种情形下，你可以配置成点到多点网络。不相邻的路由设备之间可以通过虚链路交换路由信息。

OSPF 点到多点接口可以定义成多个点到点网络接口，它建立多个主机路由。OSPF 点到多点网络与非广播、多访问网络以及点到点网络相比，有以下优点：

点到多点网络容易配置，它不要求邻居配置命令，且不需产生 DR。

因为它不要求全网状拓扑，所以它的开销更小。

它更加可靠。即使在虚连路失败的情形下，也能保持连接。

在接口配置模式下，用下面的命令配置 OSPF 的网络类型。

命令	目的
ip ospf network {broadcast | non-broadcast | {point-to-multipoint [non-broadcast] }}	这个命令配置OSPF的网络类型。

在本章末，你可以看到一个 OSPF 的点到多点的配置例子。

6.3.4 配置点到多点、广播网络

在点到多点网络、广播网络，不必描述邻居关系。然而可以用 neighbor 命令描述到某一邻居的权值。

在使用这个命令之前，一些 OSPF 点到多点协议流量为多播流量。所以，在点到多点接口，不需 neighbor 命令。Hello 包、更新包与确认包都通过广播方式发送出去，特别是，多播 hello 包能动态发现所有邻居。

在点到多点网络，路由设备假设所有的邻居的权值相等。这个权值可以通过命令 ip ospf cost 来配置。 实际上，每个邻居的带宽不一样，所以权值应不一样。这个特征仅仅用在点到多点接口。

使用下面的命令配置接口为点到多点接口且为每个邻居分配一个权值：

命令	目的
ip ospf network point-to-multipoint	在广播媒体上，配置接口为点到多点网络。
exit	进入全局配置模式。
router ospf *process-id*	配置一个OSPF 路由进程且进入路由配置模式。
neighbor ip-address **cost** number	指定一个邻居且为它分配一个权值。
neighbor ip-address **cost** number	为每一个希望指定权值的邻居，重复上述配置命令。否则邻居的权值使用命令ip ospf cost 命令指定的权值。

6.3.5 配置非广播类型网络

由于在 OSPF 网中有多个路由设备，所以必须为网络选举一个 DR。如果广播能力未被配置，则要求为选举过程进行参数配置。

这些参数仅仅在有可能成为 DR 或 BDR 的路由设备上进行配置。

在路由配置模式下，使用下面的命令配置互联非广播网络的路由设备：

命令	目的

neighbor *ip-address* [**priority** *number*] [**poll-interval** *seconds*]	配置连接到非广播网络上的路由设备。

你能指定下面的邻居路由设备参数：

- 邻居路由设备的优先级。
- 非广播 poll 间隔
- 可达邻居的接口

在点到多点、非广播网络，你能使用 neighbor 命令指定邻居关系。分配一个可选的权值。

在以前的软件版本中，一些用户在非广播媒体上，配置点到多点连接（IP over ATM），因此路由设备不能动态发现它的邻居路由设备。这个特征允许 neighbor 命令可以用在点到多点接口。

在点到多点网络，路由设备假设所有的邻居的权值相等。这个权值可以通过命令 ip ospf cost 来配置。 实际上，每个邻居的带宽不一样，所以权值应不一样。这个特征仅仅用在点到多点接口。

在接口配置模式下，使用下面的命令在不支持广播的媒体上配置点到多点接口。

命令	目的
ip ospf network point-to-multipoint non-broadcast	在非广播媒体上配点到多点接口。
exit	进入全局配置模式。
router ospf *process-id*	创建一个OSPF 路由进程且进入路由配置模式。
neighbor *ip-address* [**cost** *number*]	指定一个OSPF邻居且为它分配一个权值。
neighbor *ip-address* [**cost** *number*]	为每一个邻居重复上述配置命令。

6.3.6 配置 OSPF 域参数

可以配置的区域参数有：认证、指定 Stub 区、为默认汇总路由指定权值。认证采用基于口令保护。

Stub 区域即不分发外部路由到该区的区域。取而代之的是,在 ABR 生成一条默认外部路由进入 stub 区域，使它能到达自治区域的外部网络。为了利用 OSPF Stub 支持的特性，在 Stub 区域必须使用默认路由，为了进一步减少发送进入 Stub 区域的 LSA 数，你能在 ABR 禁止汇总（No Summary）来减少发送汇总 LSA(类型 3)进入 Stub 区域。

在路由配置模式，使用下面的命令设定区域参数：

命令	目的
area *area-id* **authentication simple**	激活OSPF区域认证。
area *area-id* **authentication**	使OSPF使用 MD5进行认证。

message-digest	
area *area-id* **stub [no-summary]**	定义一个stub区。
area *area-id* **default-cost** *cost*	为Stub区域的默认路由设定权值。

6.3.7 配置 OSPF 的 NSSA 域

NSSA 区域类似于 STUB 区域，但它能有限制的输入外部路由，通过分发的方式，NSSA 允许输入外部路由，传输中支持路由汇总和包过滤。如果 ISP 需要使用 OSPF 连接中心的网络到使用不同路由协议的远端网络时，使用 NSSA 可以简化管理。

在 NSSA 以前，企业中心边界路由设备与远端路由设备的连接不能运行在 OSPF 的 STUB 区域，因为远端网络的路由不能分发进 STUB 区域。而简单的路由协议如 RIP 可以发布，但这需要维护两种路由协议。使用了 NSSA 将中心路由设备和远端路由设备放在同一个 NSSA 区域，可以使 OSPF 延伸到远端网络。

使用 NSSA 区域的同时应该注意：一旦配置了 NSSA，该区域的 ABR 路由设备生成默认路由进入 NSSA。另外，在同一个区域的每个路由设备都必须承认该区为 NSSA 区，否则，路由设备间不能进行通信。在 ABR 上应注意使用显示的发布，避免引起在该路由设备传输包的混淆。

在路由配置模式使用下面的命令设定 OSPF 的 NSSA 区域参数。

命令	目的
Area *area-id* **nssa** [**no-redistribution**][**no-summary**][**default-information -originate**] [**translate-always**]	配置ospf nssa区域。

6.3.8 配置 OSPF 域内路由的汇总

这个特性使得 ABR 广播一条汇总路由到其他区域。在 OSPF 中，ABR 将广播每一个网络到其他区域。如果网络号按照某种方式分配，使得它们连续，你能配置 ABR 广播一条汇总路由到其他区。汇总路由能覆盖一定范围的所有网络。

在路由配置模式，使用下面的命令设定地址范围：

命令	目的
area area-id **range** address mask	设定汇总路由的地址范围。

6.3.9 配置转发路由的汇总

当从其他路由区域分发路由到 OSPF 路由区域时，每条路由以外部 LSA 的方式进行单独广播。然而你能配置路由设备广播一条路由，它能覆盖一定的地址范围。这种方式可以减少 0SPF 链路状态数据库的大小。

在路由配置模式，使用下面的命令，配置汇总路由：

命令	目的

summary-address *prefix mask* [not advertise]	描述覆盖分发路由的地址与掩码，仅仅一条汇总路由被广播。

6.3.10 生成默认路由

能要求 ASBR 生成一条默认路由进入 OSPF 路由域。无论何时，你配置路由设备分发路由进入 OSPF 路由域，该路由自动变成 ASBR。然而，ASBR 默认并不生成默认路由进入 OSPF 路由域。

在路由配置模式，使用下面的命令，强制 ASBR 生成默认路由：

命令	目的
default-information originate [**always**] [**route-map** *map-name*]	强制ASBR生成默认路由进入OSPF路由域。

6.3.11 由 LOOPBACK 接口选择路由 ID

OSPF 使用配置在接口的最大 IP 地址作为它的路由设备 ID。如果与这个 IP 地址相连的接口变为 DOWN 状态，或者该 IP 地址被删除，OSPF 进程将重新计算新的路由设备 ID 并且重新从所有接口发送路由信息。

如果一个 loopback 接口配置了 IP 地址，则路由设备使用 IP 地址作为它的路由设备 ID，由于 loopback 接口永远不会 Down，所以使得路由表具有较大的稳定性。

路由设备优先选用 Loopback 接口作为路由设备 ID，同时也选择所有 Loopback 接口中最大的 IP 地址作为路由设备 ID。如果没有 Loopback 接口，则使用路由设备的最大 IP 地址。你不能指定 OSPF 使用任何特定的接口。

在全局模式下，使用下面的命令，配置 IP Loopback 接口。

命令	目的
interface loopback 0	创建一个loopback接口且进入接口配置模式。
ip address *ip-address mask*	为接口分配一个IP 地址。

6.3.12 配置 OSPF 的管理距离

管理距离是路由信息源的信任等级，如单个路由设备或一组路由设备。一般来说，管理距离是 0－255 之间的整数，值越大，信任级别越低。如果管理距离为 255，则路由信息源不被信任且应当被忽略。

OSPF 使用三类不同的管理距离：域间、域内和外部。在一个区域内的路由是域内；到其他区域的路由是区域间；其他路由协议域分发来的路由为外部。每种类型路由的默认值为 110。

在路由配置模式下，使用下面的命令，配置 OSPF 的距离值：

命令	目的

distance ospf [**intra-area** *dist1*] [inter-area *dist2*] [**external** *dist3*]	改变OSPF 域内、域间以及外部路由管理距离值。

6.3.13 配置路由计算的计时器

你能配置 OSPF 收到拓扑变化消息与开始计算 SPF 之间的时延。也能配置连续两次计算 SPF 之间的间隔。在路由配置模式，使用下面的命令进行配置：

命令	目的
timers delay *delaytime*	设定在一个域中路由计算的时间延迟。
timers hold *holdtime*	设定在一个域中路由计算的最小时间间隔。

6.3.14 启动 On-Demand 链路的配置

OSPF 按需拨号（ospf over on demand ciruits）是对 OSPF 协议的一种改进，它使得协议在 ISDN、X.25 和拨号等按需拨号上网的情况下更为有效。我们知道，OSPF 协议将定期地在相连的路由设备之间交换 HELLO 报文和链路状态广播更新报文，即使报文的信息没有任何变化，支持这种特性后，将不再定期的传送 HELLO 报文和链路状况广播更新报文，只有在第一次建立连接和报文中包含的信息改变时才进行报文交换，这意味着，只有拓朴结构真正变化时，才会重新计算最小生成树，并进行报文传输。

如果路由设备之间是点到点连接，只需在一端配置，当然，另一端的路由设备必须支持这一特性。如果路由设备之间是点到多点的连接，只需在多点一端配置，则每一对多点连接都可以运行按需拨号；如果在单点的一端配置，则每次只能有一对点到点连接运行。

建议在 Stub 区域内配置按需拨号，只需 Stub 区域内的每台路由设备配置这种属性，区域外的路由设备可以不支持按需拨号。如果在一个标准区域配置了按需拨号，其它的所有的标准区域必须支持这种属性，因为第二类的外部链路状态广播报文将在所有的区域内传播。

在基于广播的网络上配置这种属性时，链路状态广播报文可以被抑制，但 HELLO 报文不能被抑制。因为 HELLO 报文用来维持邻居关系和选举 DR 。

该命令应在接口模式下执行：

命令	目的
ip ospf demand-circuit	配置OSPF按需拨号。

6.3.15 监视和维护 OSPF

能显示网络的统计信息，如：IP 路由表的内容、缓冲和数据库等数据。这些信息能帮助你判断网络资源的利用，解决网络问题。能了解网络节点的可达性，发现网络数据包经过网络的路由。

使用下面的命令，可以显示各种路由统计信息：

命令	目的

show ip ospf [*process-id*]	显示OSPF 路由进程的一般信息。
show ip ospf [*process-id*] **database**[**router**| **network**| **summary**| **asbr-summary**| **external**| **database-summary**]{ *link-state-id*| **self-originate**| **adv-router** [*ip-address*]}	显示OSPF数据库的相关信息。
show ip ospf border-routers	显示ABR与ASBR的内部路由表项。
show ip ospf interface	显示有关OSPF接口的信息。
show ip ospf neighbor	按照接口，显示OSPF的邻居信息。
debug ip ospf adj	监视OSPF的邻接建立过程。
debug ip ospf events	监视OSPF的接口和邻居事件。
debug ip ospf flood	监视OSPF的数据库的扩散过程。
debug ip ospf lsa-generation	监视OSPF的LSA的生成过程。
debug ip ospf packet	监视OSPF的报文。
debug ip ospf retransmission	监视OSPF的报文重发过程。
debug ip ospf spf{ **intra**| **inter**| **external**}	监视OSPF的SPF计算路由。
debug ip ospf tree	监视OSPF的SPF树的建立。

6.4 OSPF配置举例

6.4.1 OSPF 点到多点、非广播配置举例

```
Switch A:
interface vlan 1
 ip address 10.0.1.1 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 1
 network 10.0.1.0 255.0.0.0 area 0
 neighbor 10.0.1.3 cost 5
 neighbor 10.0.1.4 cost 10
```

Switch B:

```
interface vlan 1
 ip address 10.0.1.3 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 1
 network 10.0.1.0 255.0.0.0 area 0
 neighbor 10.0.1.1
 neighbor 10.0.1.4 cost 14
```

Switch C:

```
interface vlan 1
 ip address 10.0.1.4 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 1
 network 10.0.1.0 255.0.0.0 area 0
 neighbor 10.0.1.1
 neighbor 10.0.1.3
```

6.4.2 可变长子网掩码的配置举例

OSPF、静态路由支持可变长子网掩码(VLSMs). 通过 VLSMs，能在不同的接口对于同一网络号码使用不同的掩码，这节约了 IP 地址，更加有效利用网络的地址空间。

在下面的例子中，使用了 30 位的子网掩码，保留两位地址空间作为串行口的主机地址。这对于点到点的串行链路仅仅只需两个主机地址，已经足够了。

```
interface vlan 1
 ip address 131.107.1.1 255.255.255.0
!
interface serial 1/1
 ip address 131.107.254.1 255.255.255.252
!
router ospf 107
 network 131.107.0.0 255.255.0.0 area 0.0.0.0
```

6.4.3 OSPF 路由及路由分发的配置举例

OSPF 要求在众多内部路由设备、ABR 与 ASBR 间交换信息。在最小配置下，基于 OSPF 的路由设备可以在默认参数下工作，没有认证要求。

下面是三个配置例子：

- 第一个例子练习基本 OSPF 命令。
- 第二个例子练习配置虚拟链路。
- 第三个例子说明了一个更加复杂的利用 OSPF 的各种工具的配置例子。

1. 基本 OSPF 配置例子

下面的例子说明一个简单的 OSPF 配置。激活路由进程 90，连接 vlan 1 到区域 0.0.0.0。同时分发 RIP 到 OSPF，OSPF 到 RIP。

```
interface vlan 1
 ip address 130.130.1.1 255.255.255.0
```

```
!
router ospf 90
 network 130.130.0.0 255.255.0.0 area 0
 redistribute rip 1
!
router rip 1
 redistribute ospf 90
```

2. 配置内部路由设备、ABR 与 ASBR 的基本配置例子

下面的例子为四个 IP 地址范围分配了四个区域 ID。首先路由进程 109 被激活，四个区域为：10.9.50.0, 2, 3, 与 0。区域 10.9.50.0, 2, 与 3 的掩码指定了地址范围，而区域 0 包含所有的网络。

```
router ospf 109
 network 131.108.20.0 255.255.255.0 area 10.9.50.0
 network 131.108.0.0 255.255.0.0 area 2
 network 131.109.10.0 255.255.255.0 area 3
 network 0.0.0.0 0.0.0.0 area 0
 redistribute static
!
interface vlan 1
 ip address 131.108.20.5 255.255.255.0
!
interface vlan 2
 ip address 131.108.1.5 255.255.255.0
!
interface vlan 3
ip address 131.108.2.5 255.255.255.0
!
interface vlan 4
 ip address 131.109.10.5 255.255.255.0
!
interface vlan 5
 ip address 131.109.1.1 255.255.255.0
!
interface vlan 6
 ip address 10.1.0.1 255.255.0.0
!
ip route 44.0.0.0 255.0.0.0 VLAN1
!
```

网络区域配置命令作用是有顺序的，所以命令的顺序是重要的。路由设备是按顺序匹配每个接口的地址/掩码对。具体可以参见“OSPF 命令”中有关网络协议命令参考中的相关内容。

可以看看第一个网络区域。即区域 ID 10.9.50.0 配置的接口子网为 131.108.20.0。则以 vlan 1 匹配。则 vlan 1 仅在区域 10.9.50.0。

接下来第二个区域。除了 vlan 1，同样的过程分析其他接口，则 vlan 2 匹配。所以 vlan2 连接到区域 2。

继续匹配其他的网络区域。注意最后一个网络区域命令是一个特例，它表示剩下的接口都连接到网络区域 0。

3. 虚拟链路的配置例子

图 5-2 是配置例子的网络拓扑图。

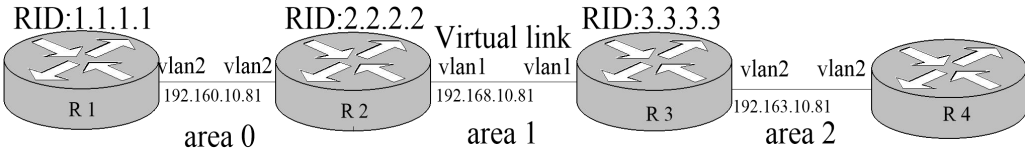


图 5-2 配置例子的网络拓扑图

按照上图配置路由设备：

R1：

```
interface vlan 2
 ip address 192.160.10.81 255.255.255.0
!
router ospf 1
 router-id 1.1.1.1
 network 192.160.10.81 255.255.255.0 area 0
!
```

R2：

```
interface vlan 1
 ip address 192.168.10.81 255.255.255.0
!
interface vlan 2
 ip address 192.160.10.82 255.255.255.0
!
router ospf 192
 router-id 2.2.2.2
 network 192.168.10.81 255.255.255.0 area 1
 network 192.160.10.82 255.255.255.0 area 0
 area 1 virtual-link 3.3.3.3
!
```

R3：

```
interface vlan 1
 ip address 192.168.10.82 255.255.255.0
```

```
!
interface vlan 2
 ip address 192.163.10.81 255.255.255.0
!
router ospf 192
 router-id 3.3.3.3
 network 192.168.10.82 255.255.255.0 area 1
 network 192.163.10.81 255.255.255.0 area 2
 area 1 virtual-link 2.2.2.2
!
```

4. 在 ABR 路由设备上配置复杂 OSPF

下面例子说明了配置 ABR 涉及的几个任务。可以分成以下两个目录：

- 基本 OSPF 配置
- 路由分发

在这个配置中的任务下面作了简单的描述。图 5-3 说明了网络地址的范围与区域的分配。

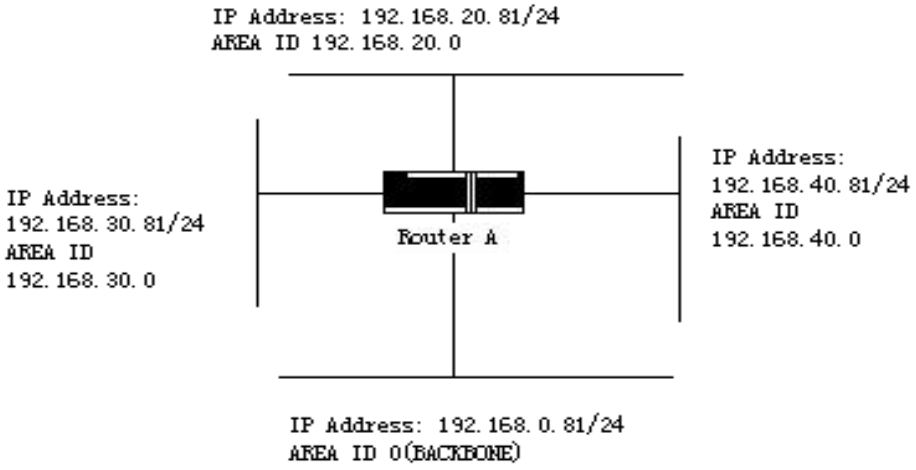


图 5-3 网络地址的范围与区域的分配

这个例子的基本配置任务如下：

- 为 vlan 1 到 4 配置地址范围。
- 激活 OSPF 在每一个接口。
- 为每个区域和网络设置 OSPF 认证口令。
- 设定链路状态权值和其他接口参数。
- 在创建 Stub 区 36.0.0.0。（注意：对于认证与 Stub 区域参数的设定，分别使用一条 area 命令。也可以用一条命令来设定这些参数。）

- 设定骨干区域 (Area 0).

与分发相连的配置任务如下：

- 分发 IGRP 与 RIP 路由进入 OSPF 的参数设置（包括 metric-type, metric, tag, 与 subnet）。
- 分发 IGRP 与 OSPF 的路由进入 RIP。

下面是 OSPF 配置例子：

```
interface vlan 1
 ip address 192.168.20.81 255.255.255.0
 ip ospf password GHGHGHG
 ip ospf cost 10
!
interface vlan 2
 ip address 192.168.30.81 255.255.255.0
 ip ospf password ijklmnop
 ip ospf cost 20
 ip ospf retransmit-interval 10
 ip ospf transmit-delay 2
 ip ospf priority 4
!
interface vlan 3
 ip address 192.168.40.81 255.255.255.0
 ip ospf password abcdefgh
 ip ospf cost 10
!
interface vlan 4
 ip address 192.168.0.81 255.255.255.0
 ip ospf password ijklmnop
 ip ospf cost 20
 ip ospf dead-interval 80
!
router ospf 192
 network 192.168.0.0 255.255.255.0 area 0
 network 192.168.20.0 255.255.255.0 area 192.168.20.0
 network 192.168.30.0 255.255.255.0 area 192.168.30.0
 network 192.168.40.0 255.255.255.0 area 192.168.40.0
 area 0 authentication simple
 area 192.168.20.0 stub
 area 192.168.20.0 authentication simple
 area 192.168.20.0 default-cost 20
 area 192.168.20.0 authentication simple
 area 192.168.20.0 range 36.0.0.0 255.0.0.0
 area 192.168.30.0 range 192.42.110.0 255.255.255.0
 area 0 range 130.0.0.0 255.0.0.0
```

```
 area 0 range 141.0.0.0 255.0.0.0
 redistribute rip 1
```

RIP 在网络 192.168.30.0：

```
router rip 1
 redistribute ospf 192
!
```

第 7 章　BGP 配置

7.1　概述

BGP 是在 RFC1163、1267 和 1771 中定义的一种外部网关协议 (EGP)。它允许建立一种自治系统间路由选择机制，该机制能自动地保证在自治系统之间进行无环路的路由选择信息交换。

7.1.1　交换机的 BGP 实现

在 BGP 中，每条路由都包含一个网络号、该路由传递所通过的自治系统列表(称作自治系统路径 as-path)、以及其他属性列表。本公司交换机软件支持 RFC1771 中定义的 BGP 版本 4。BGP 的基本功能是同其它 BGP 系统交换网络可达性信息，包括有关 AS 路径表的信息。该信息能够用于构造能消除路由环路的 AS 连通图，并且能用 AS 连通图实施 AS 级的路由策略。BGP 版本 4 支持无类型域间路由（CIDR），CIDR 通过创建汇总路由减少路由表的大小，从而产生了超网。CIDR 消除了 BGP 内网络等级的概念，并支持 IP 前缀广播。CIDR 路由能通过 OSPF, 增强型 IGRP, ISIS-IP 和 RIP2 来传送。

外部网关路由与内部网关路由一个很大区别就是可控制性更好。为了对路由进行控制，BGP 的实现提供了多种可选方法：

- 为了过滤路由，可以用基于邻居的 access-list、as-path access-list、prefix-list 来过滤路由，也可以用基于端口的 access-list、prefix-list 来过滤路由或路由的 Nexthop 属性。
- 为了改变路由的属性，可以用路由映像（route-map）来修改 MED、Local Preference、路由权值（Weight）等 BGP 路由的属性。
- 为了与内部网关动态路由协议（ospf、rip 等）进行交互，可以使用路由转发（redistribute），从而自动产生 BGP 的路由信息。也可以通过手工配置 network、aggregate 来产生 BGP 路由。在产生 BGP 路由的同时，可以用路由映射（route-map）来设置路由的属性。
- 为了控制系统中 BGP 路由的优先级，可以用 distance 命令来设置 BGP 路由的管理距离。

7.1.2　BGP 如何选择路径

BGP 的决策过程是建立在比较路由属性值的基础上的。当到同一网络具有多条路由时，BGP 选择到达目的的最佳路由。下面的过程总结了 BGP 如何选择最佳路由：

(1)　如果下一跳到达不了，就不考虑它。

(2)　如果路径是内部的，并且激活了同步，如果路由不在 IGP 中，就不考虑该路由。

(3)　优先选取具有最大权值的路径。

(4) 如果各路由有相同的权值，优先选取具有最大本地优先级的路由。

(5) 如果各路由具有相同的本地优先级值，优先选取由本地交换机产生的路由。比如，路由可能由本地交换机使用 network、aggregate 命令或通过转发 IGP 的路由生成。

(6) 如果本地优先级值相同，或者如果没有路由由本地交换机生成，则优先选取具有最短 AS 路径的路由。

(7) 如果 AS 路径长度相同，则优先选取具有最低 Origin 属性值（IGP < EGP < INCOMPLETE）的路由。

(8) 如果 Origin 属性值相同，则优先选取具有最低 MED 值的路由。除非激活 bgp always-compare-med ，否则这种比较只在来自同一邻居 AS 的路由间进行。

(9) 如果各路由具有相同的 MED，优先选取外部路径（EBGP）而不是内部路径（IBGP 在内部）。所有的自治系统联盟内部的路径都被认为是内部路径，但是优先选取联盟 EBGP 而不是联盟 IBGP。

(10) 如果各路由有相同的连接属性，优先选取具有较小 router-id 的路由。

7.2 BGP配置任务列表

BGP 配置任务分为基本任务和高级任务。基本任务中的前两项是配置 BGP 所必需的，基本任务中的其他项以及高级任务则是可选的。

7.2.1 BGP 基本配置任务

- 激活 BGP 路由选择
- 配置 BGP 邻居
- 配置 BGP 软重配置
- 复位 BGP 连接
- 配置 BGP 与 IGPs 的同步
- 配置 BGP 路由权重
- 配置基于邻居进行 BGP 路由过滤
- 配置基于端口进行 BGP 路由过滤
- 取消 BGP 更新下一跳处理

7.2.2 高级 BGP 配置任务

高级的、可选的 BGP 配置任务如下:

- 使用路由映像过滤和修改路由更新
- 配置聚合地址
- 配置 BGP 团体属性
- 配置自治系统联盟
- 配置路由反射器
- 关闭对等体
- 配置多跳数外部对等体
- 设置 BGP 路由管理距离
- 调整 BGP 定时器
- 对来自不同 AS 的路由比较 MED
- 配置 BGP 邻居 MD5 认证
- 配置 BGP 平滑重启能力
- 配置输出路由过滤（ORF 过滤）功能

要了解用于多种 IP 路由选择协议的有关配置属性的信息，参见“配置 IP 路由独立于协议的属性”。

7.3 BGP配置任务

7.3.1 配置基本 BGP 特性

1. 激活 BGP 路由选择

为了激活 BGP 路由选择，使用全局配置模式开始的如下命令激活 BGP 路由选择：

命令	目的
router bgp autonomous-system	在交换机配置模式下，激活BGP 路由过程。
Network network-number/masklen [route-map route-map-name]	将网络标记为本地自治系统并将其列入BGP表中。

注意：

对外部网关路由协议而言，用 network 交换机配置命令配置一个 IP 网络，仅能控制哪些网络能得到通告。这恰恰同内部网关协议（IGP）相反，例如 RIP,它是用 network 命令决定发送更新到何处。

network 命令用于把 IGP 路由引入到 BGP 路由表中。交换机资源，例如已配置的 RAM，决定了能使用的 **network** 命令的上限。作为选择，可以使用 **redistribute** 命令来达到同样的效果。

2. 配置 BGP 邻居

配置 BGP 邻居是为了建立交换路由信息的对象。为了与外部世界交换路由信息，必须配置 BGP 邻居。

BGP 支持两种邻居：内部邻居（IBGP）和外部邻居（EBGP）。内部邻居在同一 AS 内；外部邻居在不同 AS 内。通常，外部邻居彼此相邻，共享一个子网。而内部邻居可能在同一 AS 的任何地方。

使用交换机配置命令配置 BGP 邻居：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **remote-as** *number*	指定一个BGP邻居。

有关配置 BGP 邻居的例子，请参阅本章后面的“BGP 邻居配置示例”一节。

3. 配置 BGP 软重配置

一般说来，BGP 邻居仅在连接建立时交换全部路由，以后只交换变化的路由。因此如果配置的路由策略有变化，为了应用于已收到的路由，必须清除 BGP 会话。清除 BGP 会话导致高速缓存无效并对网络的运作产生巨大的影响。软重配置功能使得不必清除 BGP 会话就允许配置和激活策略。因此，推荐使用软重配置，目前实现基于每个邻居来进行软重新配置。当软重新配置用于从邻居产生的入站更新时，则称之为入站软重配置当软重新配置用于发送出站更新到邻居时，则称之为出站软重配置。执行入站软重配置能使新的输入策略生效，执行出站软重配置使新的本地输出策略不复位 BGP 会话就生效

为了不复位 BGP 会话就产生新的入站更新，本地 BGP 会话者应不加修改的存储接收到的入站更新，而不管它是否被当前入站策略接收或拒绝。这样会很耗费内存，应当避免使用。另一方面，出站重配置没有任何额外的内存开销，因此总是有效的。可以在 BGP 会话的另一边触发出站软重配置使本地新的入站策略生效。

要允许入站软重配置，应配置 BGP 存储所有接收到的路由更新。出站软重配置不要求预配置。

使用下面的交换机配置命令配置 BGP 软重配置:

命令	目的
Neighbor {*ip-address* | *X:X::X:X* } *soft-reconfiguration* [*inbound*]	配置BGP软重配置。

如果使用 peer-group-name 参数指定 BGP 对等组，所有对等组成员继承用该命令配置的特征。

4. 复位 BGP 连接

一旦定义了两个交换机为 BGP 邻居，它们将会建立一个 BGP 连接，并交换路由选择信息。如果随后改变了 BGP 路由策略，或者其他的配置改变，则必须复位 BGP 连接以使配置变化生效。使用下面两个管理模式命令中的任意一个复位 BGP 连接：

命令	目的
clear ip bgp *	复位所有BGP连接。
clear ip bgp *address*	重建一个特定的BGP连接。

5. 配置 BGP 与 IGPs 的同步

如果允许另一个 AS 通过你的 AS 到第三个 AS 传送数据，则你的 AS 内部路由状态同它广播给其他 AS 的路由信息保持一致是十分重要的。例如，如果在你的 AS 中的所有交换机已通过 IGP 获知路由之前，你的 BGP 要广播路由，那么你的 AS 可能会收到一些交换机不能路由的信息。为了防止这种情况发生，BGP 必须一直等到 AS 内部所有 IGP 交换机已经获知了该路由信息为止，这就是 BGP 与 IGP 的同步，同步被默认激活。

在某些情况下，不必同步。如果不允许其他 AS 通过你的 AS 传送数据，或者如果你的 AS 中的所有交换机都将运行 BGP, 就可以取消同步功能。取消该特征能允许在你的 IGP 中携带较少的路由，并使得 BGP 收敛得更快。使用下面的交换机配置命令取消同步：

命令	目的
no synchronization	取消BGP和IGP之间的同步

当取消同步时，也应该使用 **clear ip bgp** 命令清除 BGP 会话。

有关 BGP 同步的一个例子，请参阅本章后面的“由邻居进行 BGP 路径过滤示例”一节。

一般说来，不希望转发所有路由给你的 IGP。通常的设计是转发一条或两条路由，并使他们成为 IGRP 中的外部路由，或者使 BGP 会话者产生一个 AS 默认路由。当从 BGP 向 IGP 转发时，只有通过 EBGP 获取来的路由才会被转发。在大多数情况下，也不想再分配你的 IGP 给 BGP，而只是用 network 交换机配置命令列出 AS 中的网络，那么你的网络就会被广播。以这种方式列出的网络称为本地网络，并使 BGP 具有 IGP 的 Origin 属性。它们必须出现在主 IP 路由表中，并且有效；例如，它们是直连路由、静态路由或通过 IGP 获知的路由。BGP 路由过程周期性地扫描主 IP 路由表以检测本地网络存在与否，并据此适当更新 BGP 路由表如果确实想 BGP 执行转发，必须非常小心，因为可能 IGP 中的路由是其他交换机通过 BGP 转发进来的，这就出现一种 BGP 把信息潜在地引入 IGP，然后又把此信息送回 BGP 的情况。反之亦然。

6. 配置 BGP 路由权重

BGP 路由权重是赋给 BGP 路由的一个数字以便能够控制路由选择过程，权重对交换机而言是本地的。权重取值范围是 0～65535。本地生成的 BGP 路由默认权重为 32768，从邻居获知的路由权重为 0。管理者可以通过改变路由的权重实施路由策略。

使用下面的交换机配置命令配置 BGP 路由权重：

命令	目的

neighbor {*ip-address* | *X:X::X:X* } **weight** *weight*	给来自一个邻居的所有路由指定一个权重。

此外，你可以通过路由映射（route-map）改变路由的权重。

7. 配置基于邻居进行 BGP 路由过滤

交换机软件的 BGP 实现中可以用四种方式过滤指定邻居的 BGP 路由：

同 **ip as-path access-list** 全局配置命令和 **neighbor filter-list** 命令一起使用 as path 列表过滤器。

命令	目的
ip as-path access-list *aspaths-list-name* {**permit | deny**} *as-regular-expression*	定义一个与BGP有关的访问表。
router bgp *autonomous-system*	进入交换机配置模式。
neighbor {*ip-address* | *X:X::X:X* } **filter-list** *aspath-list-name* {**in | out** }	建立一个BGP过滤器。

同 **ip access-list** 全局配置命令和 **neighbor distribute-list** 命令一起使用访问列表。

命令	目的
ip access-list standard *access-list-name*	定义一个访问列表。
router bgp *autonomous-system*	进入交换机配置模式。
neighbor {*ip-address* | *X:X::X:X* } **distribute-list** *access-list-name* {**in | out** }	建立一个BGP过滤器。

同 **ip prefix-list** 全局配置命令和 **neighbor prefix-list** 命令一起使用前缀列表。

命令	目的
ip prefix-list *prefixs-list-name* {**permit |deny**} A.B.C.D/n **ge** x **le** y	定义一个前缀列表。
router bgp *autonomous-system*	进入交换机配置模式。
neighbor {*ip-address* | *X:X::X:X* } **prefix-list** *prefix-list-name* {**in | out** }	建立一个BGP过滤器。

同 route-map 全局配置命令和 neighbor route-map 命令一起使用路由映射。

使用路由映射不仅可以过滤，还可以改变路由属性，其使用在以后章节描述。

基于邻居过滤路由的示例参考“基于邻居进行 BGP 路由过滤举例”。

8. 配置基于端口进行 BGP 路由过滤

配置基于端口过滤 BGP 路由可以使用访问列表、前缀列表。可以过滤路由的网络编号，也可以过滤路由的网关地址。可以指定 access-list 选项用访问列表过滤路由的网络编号，指定 prefix-list 选项用前缀列表过滤路由的网络编号，指定 gateway 选项用访问列表过滤路由的 Nexthop 属性。甚至可以同时过滤路由的网络编号和 Nexthop 属性，但 access-list 选项不能和 prefix-list 选项一起使用。指定*可以过滤所有端口上的路由。

配置基于端口过滤 BGP 路由需要在 BGP 配置模式进行如下配置：

命令	目的
filter interface 〔**in | out**〕〔**access-list** access-list-name〕 〔**prefix-list** prefix-list-name〕〔**gateway** access-list-name〕	基于端口过滤BGP路由。

基于端口过滤路由的示例参考“基于端口进行 BGP 路由过滤举例”。

9. 取消 BGP 更新下一跳处理

可以配置取消邻居 BGP 更新的下一跳处理。这可能在非广播网络（如帧中继或 X.25）中有用，在帧中继或 X.25 中，BGP 邻居可能不能直接访问同一 IP 子网中的所有其它邻居。有两种方式取消下一跳处理：

- 使用该 BGP 连接的本地 IP 地址取代出站路由的下一跳地址；
- 使用路由映像指定入站或出站路由的下一跳地址。（参见其他章节）

使用下面的交换机配置命令取消下一跳处理并使用该 BGP 连连接的本地 IP 地址取代出站路由的下一跳地址：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **next-hop-self**	BGP邻居更新时取消下一跳处理。

用这一命令来配置，将使当前交换机通告它自己作为路由的下一跳。因此，其他 BGP 邻居将把发给该网络的包发到当前交换机。这在非广播网络环境中是有用的，因为存在一条从当前交换机到指定邻居的路径。但在广播网络环境中，这就没有意义，因为这将产生不必要的额外跳数。

7.3.2 配置高级 BGP 特征

1. 使用路由映像过滤和修改路由更新

可以在每个邻居的基础上使用路由映像过滤路由更新和修改参数属性。路由映像既可以用于入站更新又可以用于出站更新。只有通过路由映像的路由才在发送路由更新或接收路由更新时处理。

路由映像支持入站和入站更新基于 AS 路径、团体和网络编号的匹配。AS 匹配要求使用 **as-path access-list** 命令；基于团体的匹配要求使用 **community-list** 命令，基于网络的匹配要求使用 **ip access-list** 命令。

使用下面的 BGP 配置命令配置用路由映像过滤和修改路由更新：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **route-map** *route-map-name* **{in | out}**	把路由映像应用于入站或者出站路由。

使用路由映像过滤和修改路由更新的例子可以参考“BGP 路由映像举例”。

2. 配置聚合地址

无类型域间路由能够创建聚合路由（和超网）以使路由表最小化。可以通过再分配聚合路由到 BGP 或者通过使用下面的任务表中描述的有条件聚合属性，可以在 BGP 中配置聚合路由。如果在 BGP 表中至少有一条更具体的记录，就把聚合地址加入到 BGP 表中。

使用下面的一个或者多个交换机配置命令在路由表中创建聚合地址：

命令	目的
aggregate *network/len*	在BGP路由表中创建聚合地址。
aggregate *network/len* **summary-only**	只广播汇总地址。
aggregate *network/len* **attribute-map** *map-name*	通过路由映像产生指定条件的聚合地址。

有关使用 BGP 路由聚合的例子，请参阅本章后面的“BGP 路由聚合示例”一节。

3. 配置 BGP 团体属性

BGP 支持的路由策略主要基于 BGP 路由信息的以下三个值之一：

- 路由网络编号
- 路由的 AS_PATH 属性值
- 路由的 COMMUNITY 属性值

通过 COMMUNITY 属性是把路由划分为团体，并应用基于团体的路由策略，从而简化了控制路由信息的配置。

团体是一组具有共同属性的路由，每一条路由可以属于多个团体。AS 管理人员可以定义一条路由属于哪一个团体。

COMMUNITY 属性是在 1～4,294,967,200 范围内的可选的、可传递的全局属性。因特网团体中预先定义的著名团体如下：

no-export---不广播本路由到 EBGP 对等体（包括自治系统联盟内部的 EBGP 对等体）。

no-advertise---不广播本路由到任何对等体。

local-as---不广播本路由到自治系统外部（可以发送本路由到自治系统联盟中的其它子 AS 对等体。）

在生成、接收或者转发路由时，BGP 会话者可以设置、添加或者修改路由团体属性。当聚合路由时，产生的聚合包含来自所有初始路由全部团体的 COMMUNITY 属性。

默认情况下，不发送 COMMUNITY 属性到邻居。使用下面的 BGP 配置命令指定发送 COMMUNITY 属性到邻居：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **send-community**	指定发送COMMUNITY 属性到邻居。

为路由设置团体属性需要如下工作：

命令	目的
route-map *map-name sequence-number* {**deny | permit**}	配置路由映射。
set community *community-value*	配置设置规则。
router bgp *autonomous-system*	进入BGP配置模式。
neighbor{*ip-address* | *X:X::X:X* } **route-map** *access-list-name* {**in** | **out** }	应用路由映射。

基于团体属性过滤路由信息需要作如下工作：

命令	目的
ip community-list {**expanded | standard** } *community-list-name* {**permit | deny**} *communtiy-expression*	定义团体列表。
route-map *map-name sequence-number* **{deny | permit}**	配置路由映射。
match community *community-list-name*	配置匹配规则。
router bgp *autonomous-system*	进入BGP配置模式。
neighbor {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in | out** }	应用路由映射。

使用团体属性的示例可以参考“使用 BGP 团体属性的路由映像举例”。

4. 配置自治系统联盟

减少 IBGP 连接的方法是把一个 AS 分成多个子 AS，然后把它们组成一个自治系统联盟。对外界而言，联盟看起来就像一个 AS。在联盟内部，每一个子 AS 内是全连接的，并且

与同一联盟中的其他子 AS 也有连接。即使在不同子 AS 的对等体之间有 EBGP 会话，他们仍像 IBGP 对等体一样交换路由选择信息。具体地讲，是保存下一跳、MED 和本地优先信息。

要配置一个 BGP 自治系统联盟，必须指定联盟标识符。联盟标示符是一个 AS 编号，对外界而言，联盟就像一个以联盟标识符作为 AS 编号的单一 AS。

使用下面的 BGP 配置命令配置自治系统联盟标识符：

命令	目的
bgp confederation identifier *autonomous-system*	配置自治系统联盟标识符。

要指定属于自治系统联盟的自治系统号码，使用下面的 BGP 配置命令：

命令	目的
bgp confederation peers *autonomous-system* [*autonomous-system ...*]	指定属于自治系统联盟的AS。

自治系统联盟的示例请参考“BGP 自治系统联盟举例”。

5. 配置路由反射器

取代配置自治系统联盟的另一种减少 IBGP 连接的方法是配置路由反射器。

路由反射器的内部对等体被分成两组：客户对等体和 AS 中的所有其它交换机（非客户对等体）。路由反射器反射两组之间的路由，该路由反射器及其客户对等体形成一个簇。非客户对等体必须是彼此全连接的，但客户对等体不必是全连接的。簇中的客户不同簇外的 IBGP 会话者通信。

当路由反射器收到路由信息时，它就完成以下的任务：

- 广播来自外部 BGP 会话者的路由到所有客户和非客户对等体。
- 广播来自非客户的路由到所有客户。
- 广播来自客户的路由到所有客户和非客户对等体。因此，客户对等体不必是全连接的。

使用下面的交换机配置命令配置本地路由为反射器并指定邻居为路由反射器客户：

命令	目的
Neighbor {*ip-address* | *X:X::X:X* } **route-reflector-client**	配置本地交换机为路由反射器和指定邻居为客户。

一个 AS 可有多个路由反射器，路由反射器处理其它路由反射器就像处理 IBGP 会晤者一样。 通常一簇客户机只有一个路由反射器，此时该簇用路由反射器的交换机 ID 来标识。为了增加冗余度和避免单个节点的失败，一个簇可能有不止一个路由反射器。这种情况下，簇中所有的路由反射器必须用 4 字节簇 ID 来配置，以便路由反射器能够识别在同一簇中的路由反射器的更新信息。服务于同一簇的所有路由反射器应该是全连接的，并且它们应该有相同的客户和非客户对等体集合。

如果簇中有不止一个路由反射器，则是用下面的 BGP 配置命令配置簇 ID：

命令	目的
bgp cluster-id *cluster-id*	配置簇ID。

路由反射器的配置举例参见“BGP 路由反射器配置举例”。

6. 关闭对等体

使用下面的 BGP 配置命令关闭 BGP 邻居：

命令	目的
Neighbor {*ip-address* | *X:X::X:X* } **shutdown**	关闭BGP邻居。

使用下面的 BGP 配置命令激活以前关闭的邻居：

命令	目的
no neighbor {*ip-address* | *X:X::X:X* } **shutdown**	激活BGP 邻居。

7. 配置多跳数外部对等体

缺省情况下，外部对等体必须是在直接相连的网络上的，为了能配置多跳数外部对等体，需要进行如下工作：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **ebgp-multihop** ttl	配置BGP邻居为多跳数外部对等体。

8. 设置 BGP 路由管理距离

管理距离是不同路由协议优先程度的一种度量。BGP 使用三种不同的管理距离：外部距离，内部距离和本地距离。通过外部 BGP 获知的路由给出外部距离，通过内部 BGP 的路由给出内部距离，本地路由给出本地距离。使用下面的 BGP 配置命令设置 BGP 路由管理距离：

命令	目的
distance bgp {*external-distance*| *internal-distance* |*local-distance*}	设置BGP路由管理距离。

改变 BGP 路由的管理距离是危险的，一般不推荐。外部距离应该比其它任何动态路由协议的距离小，内部距离应该比其它任何动态路由协议的距离大。

9. 调整 BGP 定时器

使用下面的 BGP 配置命令调整具体邻居的 BGP **keepalive** 和 **holdtime** 定时器：

命令	目的

neighbor {*ip-address* | *X:X::X:X* } **timers** *keepalive holdtime*	为指定的对等体或者对等组设置**keepalive**和**holdtime** 定时器（以秒计）。

使用 **no neighbor timers** 命令恢复 BGP 邻居或者对等组的定时器为缺省值。

10. 对来自不同 AS 的路由比较 MED

MED 是在多个可选择的路径中选择最佳路由时考虑的一个参数。具有较低 MED 值的路径比具有较高 MED 值的路由优先考虑。

缺省情况下，在选择最佳路由过程中，MED 比较只在来自同一 AS 的路由中进行。可以允许 MEDs 比较在路由选择中进行，而不管来自于哪一个 AS 的路由。

使用下面的 BGP 配置命令来达到上述目的：

命令	目的
bgp always-compare-med	允许对来自不同AS的路由进行MEDs比较。

11. 配置 BGP 邻居 MD5 认证

为保证自治系统间路由信息的安全传递，可以通过 TCP 提供的 MD5 选项对 BGP 连接进行密码认证。

使用下面的 BGP 配置命令来达到上述目的：

命令	目的
neighbor *A.B.C.D* **password** *LINE*	启动BGP邻居的MD5认证，并设置密码。

使用 no neighbor A.B.C.D password 命令取消 BGP 邻居的 MD5 认证配置。

12. 配置 BGP 平滑重启能力

缺省情况下，BGP 协议重启时 BGP 不会保留任何学到的路由信息。

在支持 BGP GR 能力（需 BGP 会话两端都支持）以后，BGP 协议重启时 GR 能力老化路由，并保留这些老化路由，这些老化路由和普通路由没有任何区别（保留转发状态），只是打上了 stale 标记。在 BGP GR 重启过程中实现了无中断转发。

对于 Restarting Speaker（协议重启交换机），协议重启时 BGP 进程将所有 Loc-RIB 老化。一但 Restarting Speaker 与 Receiving Speaker 重新建立 BGP 会话，Restarting Speaker 将会接收并处理所有对等体发过来的 BGP 消息。可是 Restarting Speaker 会延迟路由选择过程，除非等到所有对等体的 End-of-RIB (除接收到 R 位置 1 的 Peer，以及不向外宣告 GR 能力的 Peer 的所有 Peer)，或者 update-delay timer 超时（详见 bgp update-delay）。BGP Speaker 执行路由选择，更新转发状态，老化路由标记被删除，最优路由和 End-of-RIB 被宣告给对等体。

对于 Receiving Speaker，当 Receiving Speaker 检测到拥有 GR 能力的 BGP 会话的 TCP 连接重置后，保留并老化所有从该对等体收到的相关地址簇路由信息。当出现连续重启的情况，路由会删除带的老化标记的路由信息。老化路由与其它路由信息没有什么不同，

只是带上老化标记。当与 Restarting Speaker bgp 会话建立后，Receiving Speaker 发送更新信息给 Restarting Speaker，完成路由初始更新后，发送 End-of-RIB；Receiving Speaker 收到来自对等体的路由更新，则更新老化路由并删除老标记，一旦收到相关地址簇的 End-of-RIB，Receiving Speaker 会马上删除老化标记。如果会话在“Restart Time”时间内无法重新建立 BGP 会话，Receiving Speaker 则删除带有老化标记的路由。

当 BGP 会话在 Restart Time 时间内已经建立，根据以下情况，老化路由被删除

- 会话建立后，没有收到 GR 报文
- 新接收的 GR 报文里，指定地址簇的 F 位没被置 1
- 新接收的 GR 报文里，没有该地址簇信息
- Stalepath Timer 超时

使用下面的 BGP 配置命令来达到上述目的：

命令	目的
bgp graceful-restart [**restart-time** *value*] | [**stalepath-time** *value*]	配置bgp graceful restart 能力。

13. 配置输出路由过滤（ORF 过滤）功能

输出路由过滤(ORF)是一个基于前缀的 BGP 特性，通过有 ORF 能力的通告来打开 ORF 特征。这个 ORF 能力的通告表明 BGP 会把邻居的前缀列表实施到本地配置的 ORF 上。启动 ORF 过滤功能，BGP 会安装一个入站列表来来作为一个出站列表来过滤远端对等体路由，以减少不想要的路由更新。

使用下面的 BGP 配置命令来达到上述目的：

命令	目的
neighbor {*ip-address* | *X:X::X:X* } **capability orf prefix-list**	允许ORF特性。
neighbor{*ip-address* | *X:X::X:X* } **prefix-list** *prefix-name* **in**	配置入站前缀列表，以支持ORF的ROUTER-REFRESH报文通知邻居加入出战列表。
clear ip bgp in prefix-filter	ORF入站软重置

7.4 监视和维护BGP

管理者可以显示、删除 BGP 中路由表或其他数据库的内容，也可以显示具体的统计信息值。以下部分描述了这些任务。

- 清除 BGP 路由表和数据库

- 显示路由表和系统统计信息
- 跟踪 BGP 信息

7.4.1 清除 BGP 路由表和数据库

下面的表格列出了与清除高速缓存、表格或者 BGP 数据库相关的任务，在管理模式下使用这些命令：

命令	目的
clear ip bgp *	复位所有BGP连接。
clear ip bgp *as-number*	复位指定自治系统的BGP连接。
clear ip bgp *address*	复位指定邻居的BGP连接。
clear ip bgp *address* soft 〔**in**|**out**〕	清除指定邻居的入站数据库或出站数据库。
clear ip bgp *aggregates*	清除路由聚合产生的路由。
clear ip bgp *networks*	清除转发产生的路由。
clear ip bgp *redistribute*	清除network命令产生的路由。

7.4.2 显示路由表和系统统计信息

可以显示诸如 BGP 路由表、数据库内容之类的具体统计信息。所提供的信息可用于决定资源利用以及解决网络问题。也可以显示节点可达性信息。

使用下面的管理命令显示各种路由统计信息：

命令	目的
show ip bgp	显示系统中的BGP路由表。
show ip bgp prefix	显示匹配指定前缀列表的路由。
show ip bgp community	显示团体属性的统计信息。
show ip bgp regexp *regular-expression*	显示同指定正则表达式相匹配的路由。
show ip bgp *network*	显示指定BGP路由。
show ip bgp neighbors *address*	显示指定邻居的TCP和BGP连接的详细信息。
show ip bgp neighbors *[address]* [**received-routes** | **routes** | **advertised-routes**]	显示从特殊BGP邻居获知的路由。
show ip bgp paths	显示数据库中所有的BGP路径信息。
show ip bgp summary	显示所有BGP连接的状态。

7.4.3 跟踪 BGP 信息

可以通过跟踪 BGP 信息观察 BGP 连接建立的过程，路由收发的过程，从而便于定位错误，解决问题。跟踪信息的命令如下：

命令	目的
debug ip bgp	跟踪一般的BGP信息。
debug ip bgp all	跟踪所有的BGP信息。
debug ip bgp fsm	跟踪BGP状态机。
debug ip bgp keepalive	跟踪BGP的Keepalive报文。
debug ip bgp open	跟踪BGP的Open报文。
debug ip bgp update	跟踪BGP的Update报文。

7.5 BGP配置举例

以下各节提供了 BGP 配置的例子：

7.5.1 BGP 路由映像举例

下面的例子说明如何使用路由映像修改来自邻居的入站路由属性。从邻居 140.222.1.1 接收到的并且满足 ASPATH 访问列表 aaa 的任何路由之权重都设置为 200，本地优先级值设置为 250，并且被接收，其他路由均被拒绝。

```
router bgp 100
 neighbor 140.222.1.1 route-map fix-weight in
 neighbor 140.222.1.1 remote-as 1
!
route-map fix-weight 10 permit
 match as-path aaa
 set local-preference 250
 set weight 200
!
ip as-path access-list aaa permit ^690$
ip as-path access-list aaa permit ^1800
```

在下面的例子中，路由映像 freddy 的第一个条目将起始于自治系统 690 的所有路由的 MED 属性设为 127。第二个条目使不满足上述条件的路由允许被发送到邻居 1.1.1.1：

```
router bgp 100
 neighbor 1.1.1.1 route-map freddy out
!
ip as-path access-list abc permit ^690_
ip as-path access-list xyz permit .*
!
route-map freddy 10 permit
```

```
 match as-path abc
 set metric 127
!
route-map freddy 20 permit
match as-path xyz
```

下面的例子表示如何使用路由映像修改来自路由转发的路由：

```
router bgp 100
 redistribute rip 1 route-map rip2bgp
!
route-map rip2bgp
 match ip address rip
 set local-preference 25
 set metric 127
 set weight 30000
 set ip next-hop 192.92.68.24
 set origin igp
!
ip access-list standard rip
 permit 131.108.0.0 255.255.0.0
 permit 160.89.0.0 255.255.0.0
 permit 198.112.0.0 255.255.128.0
```

7.5.2 BGP 邻居配置举例

在下面的例子中，BGP 交换机属于 AS109，它生成两个网络。该交换机共有三个邻居：第一个邻居是外部邻居（在不同的 AS）；第二个邻居是内部邻居（具有相同的 AS 编号）。第三个邻居也是外部邻居。

```
router bgp 109
 network 131.108.0.0
 network 192.31.7.0
 neighbor 131.108.200.1 remote-as 167
 neighbor 131.108.234.2 remote-as 109
 neighbor 150.136.64.19 remote-as 99
```

7.5.3 基于邻居进行 BGP 路由过滤举例

下面是一个基于邻居进行 BGP 路径过滤的例子。通过 as-path 访问列表 test1 的路由将会获得权值 100。只有通过 as-path 访问列表 test2 的路由才会发送到 193.1.12.10，类似地，只有通过访问列表 test3 的路由才会被 193.1.12.10 接收：

```
router bgp 200
neighbor 193.1.12.10 remote-as 100
neighbor 193.1.12.10 filter-list test1 weight 100
neighbor 193.1.12.10 filter-list test2 out
neighbor 193.1.12.10 filter-list test3 in
!
```

```
ip as-path access-list test1 permit _109_
ip as-path access-list test2 permit _200$
ip as-path access-list test2 permit ^100$
ip as-path access-list test3 deny _690$
ip as-path access-list test3 permit .*
```

7.5.4 基于端口进行 BGP 路由过滤举例

下面是基于端口进行路由过滤的配置举例。通过访问列表 ACL 过滤从端口 vlan1 来的路由：

```
router bgp 122
filter vlan1 in access-list acl
```

下面的例子同时使用访问列表 filter-network 过滤路由的网络编号，访问列表 filter-gateway 过滤路由的网关地址过滤来自 vlan1 端口的路由：

```
router bgp 100
filter vlan1 in access-list filter-network gateway filter-gateway
```

下面的例子同时使用前缀列表 filter-prefix 过滤路由的网络编号，访问列表 filter-gateway 过滤路由的网关地址过滤来自所有端口的路由：

```
router bgp 100
 filter * in prefix-list filter-prefix gateway filter-gateway
```

7.5.5 使用前缀列表配置路由过滤举例

下面的例子拒绝缺省路由 0.0.0.0/0:

```
ip prefix-list abc deny 0.0.0.0/0
```

下面的例子允许同前缀 35.0.0.0/8 相匹配的路由

```
ip prefix-list abc permit 35.0.0.0/8
```

在下面的例子中，BGP 过程仅接收前缀长度从/8 到 /24 的前缀：

```
router bgp 1
 network 101.20.20.0
 filter * in prefix max24
!
 ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
!
```

在下面的配置中，交换机过滤从所有端口收到的路由，仅接收前缀为 8 到 24 的路由：

```
router bgp 12
filter * in prefix-list max24
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
```

下面是一些其他的前缀列表的配置例子。

下面的例子允许在网络 192/8 中前缀长度不超过 24 的路由：

ip prefix-list abc permit 192.0.0.0/8 le 24

下面的例子拒绝在网络 192/8 中前缀长度超过 25 的路由：

ip prefix-list abc deny 192.0.0.0/8 ge 25

下面的例子允许在所有地址空间中前缀长度大于 8 但小于 24 的路由：

ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24

下面的例子拒绝在所有地址空间中前缀长度大于 25 的路由

ip prefix-list abc deny 0.0.0.0/0 ge 25

本例拒绝网络 10/8 的所有路由，因为如果 A 类网 10.0.0.0/8 的掩码小于或者等于 32 位，则将拒绝其所有路由：

ip prefix-list abc deny 10.0.0.0/8 le 32

下面的例子拒绝网络 204.70.1/24 中掩码长度超过 25 的路由：

ip prefix-list abc deny 204.70.1.0/24 ge 25

下面的例子允许所有的路由：

ip prefix-list abc permit any

7.5.6 BGP 路由聚合举例

下面的例子说明如何在 BGP 中生成聚合路由，生成的方式可以是通过路由转发或者使用条件聚合路由功能。

在下面的例子中，redistribute static 命令用于转发聚合路由 193. * . * . * :

```
ip route 193.0.0.0 255.0.0.0 null 0
!
router bgp 100
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你的 AS，并且具有指示信息可能丢失的 atomic 属性。

```
router bgp 100
aggregate 193.0.0.0/8
```

下面的例子不仅创建了聚合路由 193. *. *. *，而且还抑制了把更具体的路由广播给所有邻居：

```
router bgp 100
aggregate 193.0.0.0/8 summary-only
```

7.5.7 BGP 路由反射器配置举例

下面是一个路由反射器的配置例子。RTA、RTB、RTC、RTE 属于同一自治系统 AS200，RTA 充当路由反射器，RTB、RTC 为路由反射器客户，RTE 为普通 IBGP 邻居。RTD 属于 AS100，与 RTA 建立 EBGP 连接。配置如下：

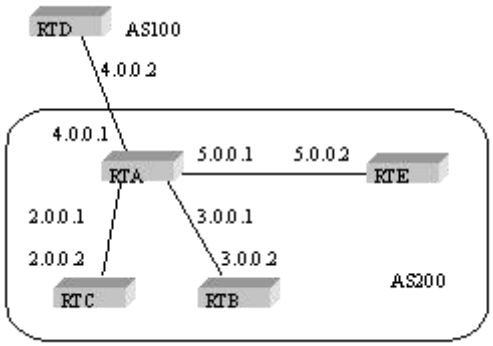


图 6-1 BGP 路由反射器配置示意图

RTA 配置：

```
interface vlan2
ip address 2.0.0.1 255.0.0.0
!
interface vlan3
ip address 3.0.0.1 255.0.0.0
!
interface vlan4
ip address 4.0.0.1 255.0.0.0
!
interface vlan5
ip address 5.0.0.1 255.0.0.0
!
router bgp 200
neighbor 2.0.0.1 remote-as 200     /*RTC IBGP*/
neighbor 2.0.0.1 route-reflector-client
neighbor 3.0.0.1 remote-as 200     /*RTB IBGP*/
neighbor 3.0.0.1 route-reflector-client
neighbor 5.0.0.1 remote-as 200     /*RTE IBGP*/
neighbor 4.0.0.2 remote-as 100     /*RTD EBGP*/
network 11.0.0.0/8
!
ip route 11.0.0.0 255.0.0.0 2.0.0.12
```

RTB 配置：

```
interface vlan3
ip address 3.0.0.2 255.0.0.0
!
router bgp 200
neighbor 3.0.0.1 remote-as 200     /*RTA IBGP*/
network 13.0.0.0/8
!
ip route 13.0.0.0 255.0.0.0 3.0.0.12
```

RTC 配置：

```
interface vlan2
ip address 2.0.0.2 255.0.0.0
!
router bgp 200
neighbor 2.0.0.1 remote-as 200     /*RTA IBGP*/
network 12.0.0.0/8
!
ip route 12.0.0.0 255.0.0.0 2.0.0.12
```

RTD 配置：

```
interface vlan4
ip address 4.0.0.2 255.0.0.0
!
router bgp 100
neighbor 4.0.0.1 remote-as 200     /*RTA EBGP*/
network 14.0.0.0/8
!
ip route 14.0.0.0 255.0.0.0 4.0.0.12
```

RTE 配置：

```
interface vlan5
ip address 5.0.0.2 255.0.0.0
!
router bgp 200
neighbor 5.0.0.1 remote-as 200     /*RTA IBGP*/
network 15.0.0.0/8
!
ip route 15.0.0.0 255.0.0.0 5.0.0.12
```

7.5.8 BGP 自治系统联盟举例

下面是一个自治系统联盟的配置，RTA、RTB、RTC 建立 IBGP 连接，属于私有自治系统 65010；RTE 属于私有自治系统 65020；RTE 与 RTA 建立自治系统联盟内部 EBGP 连接；AS65010、AS65020 组成自治系统联盟，自治系统号为 AS200；RTD 属于自治系统 AS100，RTD 通过 RTA 与自治系统 200 建立 EBGP 连接。

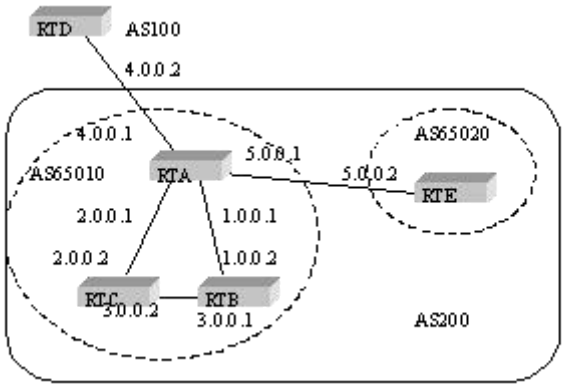


图 6-2 BGP 自治系统联盟配置示意图

RTA 配置：

```
interface vlan1
ip address 1.0.0.1 255.0.0.0
!
interface vlan2
ip address 2.0.0.1 255.0.0.0
!
interface vlan4
ip address 4.0.0.1 255.0.0.0
!
interface vlan5
ip address 5.0.0.1 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
neighbor 1.0.0.2 remote-as 65010    /*RTB IBGP*/
neighbor 2.0.0.2 remote-as 65010    /*RTC IBGP*/
neighbor 5.0.0.2 remote-as 65020    /*RTE EBGP*/
neighbor 4.0.0.2 remote-as 100      /*RTD EBGP*/
```

RTB 配置：

```
interface vlan1
ip address 1.0.0.2 255.0.0.0
!
interface vlan3
ip address 3.0.0.1 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
```

```
neighbor 1.0.0.1 remote-as 65010    /*RTA IBGP*/
neighbor 3.0.0.2 remote-as 65010    /*RTC IBGP*/
```

RTC 配置：

```
interface vlan2
ip address 2.0.0.2 255.0.0.0
!
interface vlan3
ip address 3.0.0.2 255.0.0.0
!
router bgp 65010
bgp confederation identifier 200
bgp confederation peers 65020
neighbor 2.0.0.1 remote-as 65010    /*RTA IBGP*/
neighbor 3.0.0.1 remote-as 65010    /*RTB IBGP*/
```

RTD 配置：

```
interface vlan4
ip address 4.0.0.2 255.0.0.0
!
router bgp 100
neighbor 4.0.0.1 remote-as 200    /*RTA EBGP*/
```

RTE 配置：

```
interface vlan5
ip address 5.0.0.2 255.0.0.0
!
router bgp 65020
bgp confederation identifier 200
bgp confederation peers 65010
neighbor 5.0.0.1 remote-as 65010    /*RTA EBGP*/
```

7.5.9 使用 BGP 团体属性的路由映像举例

本节包括使用具有 BGP 团体属性的路由映像的三个例子。

在第一个例子中， route map set-community 用于到邻居 171.69.232.50 的出站更新。通过访问列表 aaa 的路由设置特殊的团体属性值"no-export"，其它的路由进行正常广播。该特殊的团体属性值自动地阻止 AS200 内的 BGP 会话者将该路由广播到自治系统外。

```
router bgp 100
 neighbor 171.69.232.50 remote-as 200
 neighbor 171.69.232.50 send-community
 neighbor 171.69.232.50 route-map set-community out
!
 route-map set-community 10 permit
 match ip address aaa
 set community no-export
```

```
!
 route-map set-community 20 permit
```

在第二个例子中，route map set-community 用于到邻居 171.69.232.90 的出站更新。所有由 AS 70 产生的路由设置将团体属性值 200 200 添加到现有的值中，所有其它路由进行正常广告。

```
route-map bgp 200
 neighbor 171.69.232.90 remote-as 100
 neighbor 171.69.232.90 send-community
 neighbor 171.69.232.90 route-map set-community out
!
route-map set-community 10 permit
 match as-path test1
 set community-additive 200 200
!
route-map set-community 20 permit
 match as-path test2
!
ip as-path access-list test1 permit 70$
ip as-path access-list test2 permit .*
```

在第三个例子中，根据路由的团体属性值有选择性地设置来自邻居 171.69.232.55 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 8000，这可能包括有团体值"100 200 300" 或 "900 901."的路由。这些路由也可能有其它属性值。

所有传送团体列表 com2 的路由都设置本地优先级值为 500。

所有其他路由都设置本地优先级值为 50。因此，邻居 171.69.232.55 的所有剩余路由的本地优先值为 50。

```
router bgp 200
neighbor 171.69.232.55 remote-as 100
neighbor 171.69.232.55 route-map filter-on-community in
!
route-map filter-on-community 10 permit
match community com1
set metric 8000
!
route-map filter-on-community 20 permit
match community com2
set local-preference 500
!
route-map filter-on-community 30 permit
set local-preference 50
!
ip community-list standard com1 permit 100 200 300
ip community-list standard com1 permit 900 901
!
ip community-list standard com2 permit 88
```

```
ip community-list standard com2 permit 90
!
```

第 8 章　策略路由 PBR 配置

8.1　PBR概述

PBR 是策略路由(Policy Based Routing)的英文缩写。PBR 使得用户可以依靠某种策略来进行路由，而不是依赖路由协议。PBR 目前支持的策略是：ip 报文大小、源 ip 地址。用户可以为符合策略的报文指定下一跳 ip address 或者下一条端口。PBR 支持负载均衡，对符合策略的报文可以应用多个下一跳 ip 地址或端口。

PBR 应用下一跳的规则如下：

- 如果配置了 set ip next-hop，并且 nexthop 是可到达的，则采用 nexthop。如果有多个 nexthop，则采用第一个可到达的 nexthop，如果多个 nexthop 是 load-balance 方式，则轮流选择这些 nexthop。
- 如果配置了 set interface，并且 interface 处于可路由(端口协议 up，配置了 ip 地址)状态，则采用该端口作为下一跳端口。如果有多个 interface，则采用第一个可路由的端口，如果这些多个 interface 是 load-balance 方式，则轮流选择这些端口。如果同时配置 set ip next-hop 和 set interface，优先选择 set ip next-hop。
- set ip default next-hop 或 set default interface 仅在基于目的 ip 地址的路由表查找失败的情况下有效。

对以下报文，不会应用策略路由：

- 对于目的地址是本地的报文。
- multicast 报文
- 本地直连广播报文

8.2　PBR配置任务列表

如果想配置 PBR，有以下任务需要完成。

- 创建访问列表(可选)
- 创建 route-map
- 在端口应用策略路由
- 维护 PBR

8.3 PBR配置任务

8.3.1 创建访问列表

要创建访问列表，在全局配置模式下按以下步骤进行：

命令	说明
ip access-list standard net1	进入访问列表配置模式，定义访问列表。

8.3.2 创建 route-map

要创建 route-map，在全局配置模式下按以下步骤进行：

命令	说明
route-map pbr	进入route-map配置模式。
match ip address *access-list* **match length** *min_length max_length*	配置匹配策略。
set ip [**default**] **next-hop** *A.B.C.D* **set** [**default**] **interface** *interface_name*	配置IP报文下一跳地址或端口。

8.3.3 在端口应用策略路由

按以下步骤在 ip 报文接收端口应用策略路由：

命令	说明
interface *interface_name*	进入端口配置模式。
ip policy route-map *route-map_name*	在端口应用策略路由。

8.3.4 维护 PBR

维护 PBR，在管理态下，按以下步骤进行：

命令	说明
debug ip policy	查看应用策略路由的结果。

8.4 PBR配置举例

交换机配置：

```
!
interface Vlan1
ip address 10.1.1.3 255.255.255.0
```

```
no ip directed-broadcast
ip policy route-map pbr
!
interface Vlan2
ip address 13.1.1.3 255.255.255.0
no ip directed-broadcast
!
interface Vlan3
ip address 14.1.1.3 255.255.255.0
no ip directed-broadcast
!
ip access-list standard net1
permit 10.1.1.2 255.255.255.255
!
ip access-list standard net2
permit 10.1.1.4 255.255.255.255
!
ip access-list standard net3
permit 10.1.1.21 255.255.255.255
!
route-map pbr 10 permit
match ip address net1
set ip next-hop 13.1.1.99
!
route-map pbr 20 permit
match ip address net2
set ip next-hop 14.1.1.99
!
route-map pbr 30 permit
match ip address net3
set ip next-hop 13.1.1.99 14.1.1.99 load-balance
!
route-map pbr 40 permit
set ip default next-hop 13.1.1.99
```

配置说明

交换机将对从 vlan1 收到的报文应用策略路由。对于源 ip 地址是 10.1.1.2 的报文，其下一跳是 13.1.1.99，如果在路由表中有到下一跳 13.1.1.99 的路由，则应用该路由，如果在路由表中没有到 13.1.1.99 的路由，则采用根据目的 ip 地址查找路由表获得的路由。对于源 ip 地址是 10.1.1.21 的报文，将应用 route-map pbr 30，由于 set ip next-hop 带了 load-balance 参数，下一跳轮流选择 13.1.1.99、14.1.1.99（假设路由表中有到 13.1.1.99 和 14.1.1.99 的路由 ）。