红茶三杯
Vinsoney

BGP 基础实验手册

By 红茶三杯

http://t.sina.com/vinsoney

访问 http://ccietea.com 获得文档的最新版本

文档更新时间：2011-12

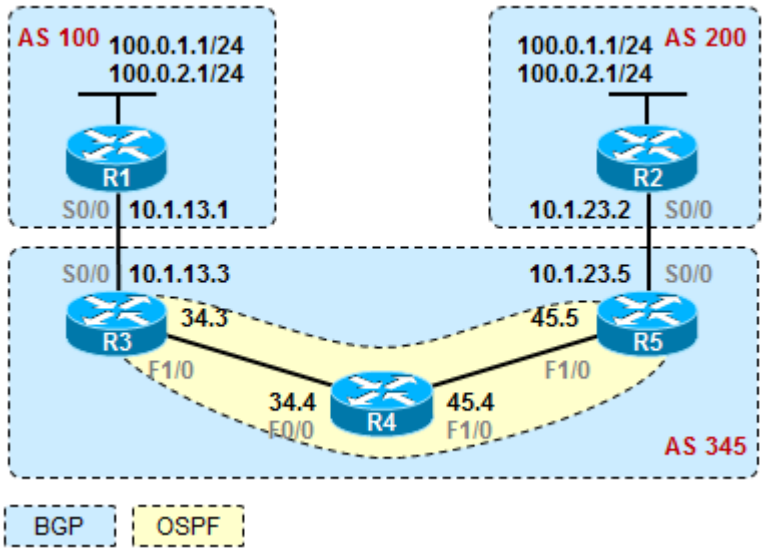
密级 ☑开放 ☐内部 ☐机密

类型 ☐讨论版 ☐测试版 ☑正式版

修订记录				
修订日期	**修订人**	**版本号**	**审核人**	**修订说明**
2011-12	**红茶三杯**			

目 录

1 实验拓扑及描述



实验描述

1. 网络拓扑及互联 IP 地址规划如图所示
2. R3、R4、R5 各自创建 LOOPBACK 接口，IP 地址为 **x.x.x.x**，x 为路由器的编号
3. R3、R4、R5 运行 OSPF，宣告三者互联接口及各自的 LOOPBACK
4. BGP 的 AS 规划如图所示
5. 完成基本的 IP、IGP 配置，建立 BGP 连接

实验需求

1. 完成基本的 BGP 配置，R1 与 R3、R2 与 R5 建立 EBGP 邻居关系；R3 与 R4、R4 与 R5 建立 IBGP 邻居关系（使用 LOOPBACK 接口作为更新源）。
2. 验证 BGP 选路规则，同时测试 BGP 相关策略工具。

2 BGP 选路规则

我们先回顾一下 BGP 的 13 条选路规则，在本实验手册中，将对选路规则中的主要条目做验证，同时也熟悉一下 BGP 的各种属性。

1. Weight 越大越优先
2. Local_Pref 越大越优先

3. 起源于本地的路由优先(如本地 network 的 ,或 aggregate 的) ,即下一跳是 0.0.0.0(在 BGP 表中,当前路由器通告的路由的下一跳为 0.0.0.0)
4. AS-Path 越短越优先
5. Origin 属性（优先顺序：IGP > EGP > Incomplete）
6. MED 越小越优先
7. 优选 EBGP 邻居发来的路由(相对于 IBGP 邻居) ,在联邦 EBGP 和 IBGP 中优选联盟 EBGP 路由
8. 优选到 BGP NEXT_HOP 最近的路由，该路由是去往下一跳路由器 IGP 度量值最小的路由
9. 如果有多条来自相同相邻 AS 的路由并通过 Maximum-paths 使多条路径可用，则将所有开销相同的路由加入 Loc-RIB
10. 如果路由都来自 EBGP 邻居，则优选最老的 EBGP 邻居传来的路由，降低滚翻的影响
11. BGP 邻居的 RID 越小越优先
12. 如果多条路径始发路由器 ID 或路由器 ID 相同，那么优选 Cluster-List 最短的路径
13. 选择邻居 ip 地址最小的路由（BGP 的 neighbor 配置中的那个邻居的地址，也就是邻居的更新源 IP）

好，那么下面我们开始实验：

3 完成基本配置

AS345 中的 R3、R4、R5 运行 OSPF。AS 内部使用 IGP 保证内部路由的互通，以满足内部的数据传输需求，同时也为 IBGP 连接提供底层路由的支持，在者作为传输 AS，”Transit AS”，运行 IGP 还能保证 BGP 路由在 AS 内的传递，而 BGP 路由的有效性(如 NEXT_HOP 属性的可达性)也需要 IGP 做一个基本的保证。所以，第一步我们先完成这个 IGP 协议的配置。

R1 的配置如下：

```
hostname R1
interface s0/0
  ip address 10.1.13.1 255.255.255.0
interface loopback1
  ip address 100.0.1.1 255.255.255.0
interface loopback2
  ip address 100.0.2.1 255.255.255.0
```

R2 的配置如下：

```
hostname R2
interface s0/0
  ip address 10.1.25.2 255.255.255.0
interface loopback1
  ip address 100.0.1.1 255.255.255.0
interface loopback2
  ip address 100.0.2.1 255.255.255.0
```

R3 的配置如下：

```
hostname R3
interface s0/0
  ip address 10.1.13.3 255.255.255.0
interface fa1/0
  ip address 10.1.34.3 255.255.255.0
interface loopback0
  ip address 3.3.3.3 255.255.255.0
router ospf 100
  network 10.1.34.0 0.0.0.255 area 0
  network 3.3.3.3 0.0.0.0 area 0
```

R4 的配置如下：

```
hostname R4
interface fa0/0
  ip address 10.1.34.4 255.255.255.0
interface fa1/0
  ip address 10.1.45.4 255.255.255.0
interface loopback0
  ip address 4.4.4.4 255.255.255.0
router ospf 100
  network 10.1.34.0 0.0.0.255 area 0
network 10.1.45.0 0.0.0.255 area 0
  network 4.4.4.4 0.0.0.0 area 0
```

R5 的配置如下：

```
hostname R5
interface s0/0
```

```
 ip address 10.1.25.5 255.255.255.0
interface fa1/0
 ip address 10.1.45.5 255.255.255.0
interface loopback0
 ip address 5.5.5.5 255.255.255.0
router ospf 100
 network 10.1.25.0 0.0.0.255 area 0
network 10.1.45.0 0.0.0.255 area 0
 network 5.5.5.5 0.0.0.0 area 0
```

如此一来，所有设备的直连网段也通了，AS345 内，路由也都通了，这是为 BGP 做准备。

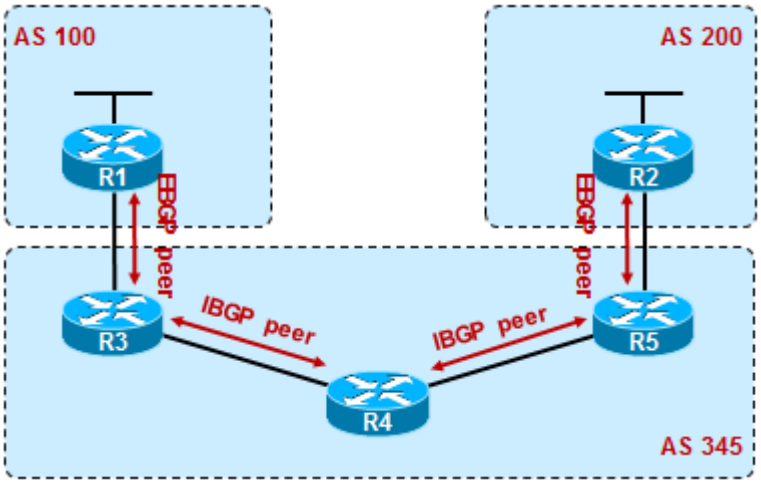
要注意我们并没有在 OSPF 中宣告 R3 的 s0/0 及 R5 的 S0/0 接口。

4 建立 BGP 邻居关系

这一步我们在上面的基础之上运行 BGP 协议，完成基本的 BGP 邻居关系的建立：

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3 及 R4、R4 及 R5 建立 IBGP 邻居关系，R3R4R5 使用 LOOPBACK 作为更新源并互指 neighbor。



R1 的配置如下：

```
router bgp 100
 no synchronization
 no auto-summary
 neighbor 10.1.13.3 remote-as 345
```

R2 的配置如下：

```
router bgp 200
  no synchronization
  no auto-summary
  neighbor 10.1.25.5 remote-as 345
```

R3 的配置如下：

```
router bgp 345
  no synchronization
  no auto-summary
  neighbor 10.1.13.1 remote-as 100          // EBGP 邻居
  neighbor 4.4.4.4 remote-as 345            // IBGP 邻居，使用 loopback0 口建邻居
  neighbor 4.4.4.4 update-source loopback 0 //  指定更新源为 loopback0
```

R4 的配置如下：

```
router bgp 345
  no synchronization
  no auto-summary
  neighbor 3.3.3.3 remote-as 345
  neighbor 3.3.3.3 update-source Loopback0
  neighbor 5.5.5.5 remote-as 345
  neighbor 5.5.5.5 update-source Loopback0
```

R5 的配置如下：

```
router bgp 345
  no synchronization
  no auto-summary
  neighbor 4.4.4.4 remote-as 345
  neighbor 4.4.4.4 update-source Loopback0
  neighbor 10.1.25.2 remote-as 200
```

注意事项：

- 一般情况下，我们会用直连接口的 IP 建立 EBGP 邻居关系。我们也通常使用 loopback 接口作为更新源、建立 IBGP 邻居关系，当我们使用 loopback 接口建立邻居关系时，务必确保本地路由器到达 IBGP 邻居的更新源接口（loopback 口）三层能通，也就是有 IBGP 邻居 LOOPBACK 接口的路由。

通过如上配置，BGP 邻居关系即可建立，我们来 show 一下：

R3#show ip bgp summary

```
BGP router identifier 3.3.3.3, local AS number 345
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
4.4.4.4         4   345      14      15        1    0    0 00:11:53        0
10.1.13.1       4   100       8       8        1    0    0 00:04:30        0
```

R3 有两个 BGP 邻居，一个是 10.1.13.1，是 EBP peer，另一个是 4.4.4.4 是 IBGP peer。

5 BGP 路由的引入及传递

R1 在 BGP 进程中宣告 LOOPBACK 网段

```
router bgp 100
  network 100.0.1.0 mask 255.255.255.0
  network 100.0.2.0 mask 255.255.255.0
```

IGP 协议，例如 OSPF，如果宣告某个直连接口（所关联的网段），则一方面该接口所对应的网络号将被引入 OSPF 路由选择进程，另一方面该接口将激活 OSPF，并开始发送 OSPF HELLO 包；而 BGP 却与此不同，BGP network 可以宣告本地接口，亦可宣告路由表中存在的路由，并且当 network 直连接口时，该接口并不会被“激活”并尝试建立邻居关系（BGP 的邻居关系需要手工 neighbor 去指的），BGP 的 network 命令，仅仅是将本地路由装载入 BGP 进程的一种方式。

在 R3 上查看 BGP 表，就能看到这两条路由

R3#show ip bgp

```
BGP table version is 3, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.1.0/24     10.1.13.1                0             0 100 i
*> 100.0.2.0/24     10.1.13.1                0             0 100 i
```

并且由于这两条学习自 R1 的 BGP 路由的 NEXT_HOP 属性为 10.1.13.1 是直连，是可达的，

因此他们都会被装进 IP 全局路由表，标记为 B。

同时从两条路由在 BGP 表中的标记：“ *> ”，说明这两条路由是 valid 可用的，且是 best 最优或者说，是优化的，因此他们会被传递给 R3 的 IBGP 邻居，也就是 R4

那么我们去 R4 上看看：

R4#sh ip bgp

```
BGP table version is 1, local router ID is 4.4.4.4
   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.1.0/24    10.1.13.1                0    100      0    100 i
* i100.0.2.0/24    10.1.13.1                0    100      0    100 i
```

我们发现 R4 虽然通过 BGP 学习到了这两条前缀，也是 valid 的(打了*号)，但是却不优化(不是 best，没有>标记)，不优化，自然也就不会加载进路由表。那这是为什么呢？ 留意到两条路由的 next hop 是 10.1.13.1，而值得注意的是，10.1.13.0 的直连网段并没有被 R3 宣告进 OSPF，换而言之 R4 去往这两条路由的 next-hop 不可达，下一跳不可达，因此这两条 BGP 路由虽然在 BGP 表中，但是不 best，也无法装载进路由表。

解决办法有：1、在 R3 上将 10.1.13.0 的直连网段宣告进 OSPF，R5 同理；2、在 R3 上修改 next-hop，将 next-hop 修改为 R3 的更新源 IP 也就是 loopback 接口 IP

R3 增加配置如下：

```
router bgp 345
  neighbor 4.4.4.4 next-hop-self
```

如此一来，R3 更新给 R4 的 BGP 路由，next-hop 将会被替换成 R3 的 loopback 接口 IP，而 R3 的 loopback 接口 IP 已被宣告进 OSPF，R4 正好通过 OSPF 学习到 loopback 路由。因此，R4 的 BGP 表如下：

```
R4#sh ip b
BGP table version is 3, local router ID is 4.4.4.4
   Network          Next Hop            Metric LocPrf Weight Path
*>i100.0.1.0/24    3.3.3.3                  0    100      0 100 i
*>i100.0.2.0/24    3.3.3.3                  0    100      0 100 i
```

我们发现两条路由都 best 了，并且都被装载进了路由表中

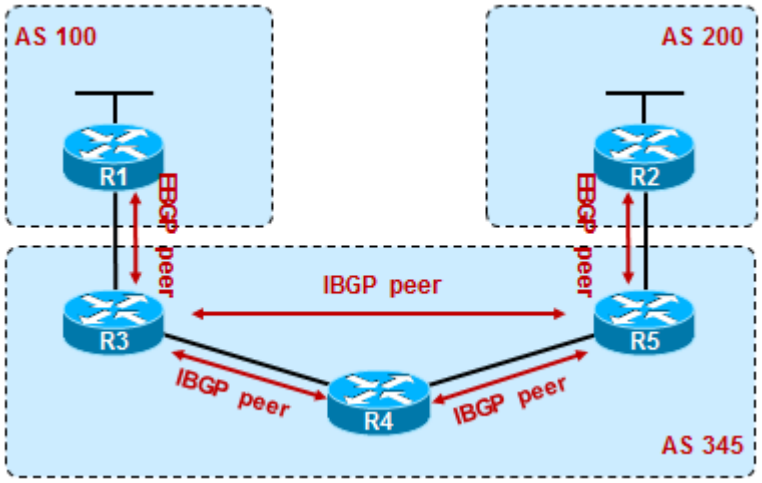
实验继续，经过上面的配置，R4 已经通过 BGP 学习到了源自 R1 的两条 BGP 路由。

但是我们却发现 R4 没有将这两条路由传递给 R5，这是因为 IBGP 的水平分割原则的效果。根据 IBGP 水平分割原则，一个 BGP 路由器，如果从它的 IBGP 邻居学习到 BGP 路由，那么它将不能把这些 BGP 路由再传递给其他 IBGP 邻居。设定这条规则的原因是，BGP 防环需要借助于

AS_PATH，而 AS_PATH 只有当路由出了本 AS 或被 BGP 路由器更新给 EBGP 邻居时才会改变，在 AS 内部是不会改变的，因此，在 AS 内部，防环需借助水平分割原则。那么如何解决 R5 学习不到路由的问题呢？

方法之一是：建立全互联 IBGP

在 R3 及 R5 之间增加一条 IBGP 连接，如此 R3 及 R5 的 BGP 路由传递就不需要借助 R4 了（虽然数据包仍需经 R4 转发，但对于 R4 而言，R3 传递给 R5 的这些数据包，只是普通的 IP 报文）。这里要留意，BGP 与 IGP 不同，IGP 如 OSPF，建立邻居关系需两台路由器直连，BGP 则无需直连，因为它是基于 TCP 建立的连接。



R3 上增加如下配置：

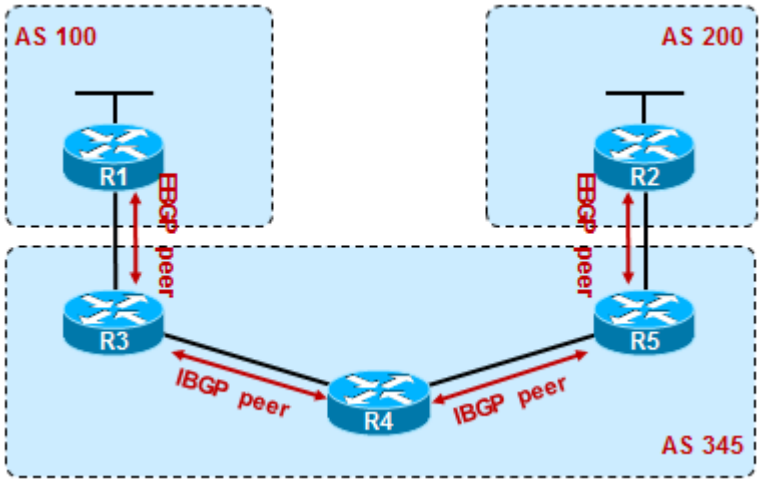
```
router bgp 345
  neighbor 5.5.5.5 remote-as 345
  neighbor 5.5.5.5 update-source Loopback0
  neighbor 5.5.5.5 next-hop-self
```

R5 上增加如下配置：

```
router bgp 345
  neighbor 3.3.3.3 remote-as 345
  neighbor 3.3.3.3 update-source Loopback0
  neighbor 3.3.3.3 next-hop-self
```

其他的解决办法，我们将在后面的实验中继续介绍

6 增加冗余路由



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系，R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

为了增加网络的可靠性及冗余性，我们让 R1 及 R2 都将 100.0.1.0 及 100.0.2.0 注入 BGP，这里是模拟冗余环境，也即我们即可通过 R1，亦可通过 R2 前往 100.0.1.0 及 2.0 网络。

R2 的配置如下：

```
router bgp 200
  no synchronization
  network 100.0.1.0 mask 255.255.255.0
  network 100.0.2.0 mask 255.255.255.0
  neighbor 10.1.25.5 remote-as 345
  no auto-summary
```

如此一来 R5 的 BGP 表就变成了如下：

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 100.0.1.0/24	10.1.25.2	0		0	200 i
* i	3.3.3.3	0	100	0	100 i
*> 100.0.2.0/24	10.1.25.2	0		0	200 i
* i	3.3.3.3	0	100	0	100 i

也就是说 R5 去往 100.0.1.及 2.0 各存在两条路径，分别是通过 10.1.25.2 及 3.3.3.3，而最终 R5 选择 10.1.25.2 作为前往这两个网段的下一跳并将路由装入路由表。那么为什么 R5 会选择 R2 呢？

而不是选择 R3 呢？这是根据 BGP 的选路原则决定的（详请见红茶三杯朱 SIR 的 BGP 文档），在这里，最终影响选路的是这么一条规则“优选 EBGP 邻居发来的路由（相对于 IBGP 邻居学过来的）”，R2 是 R5 的 EBGP 邻居，所以优选了。

我们再来看看 R4

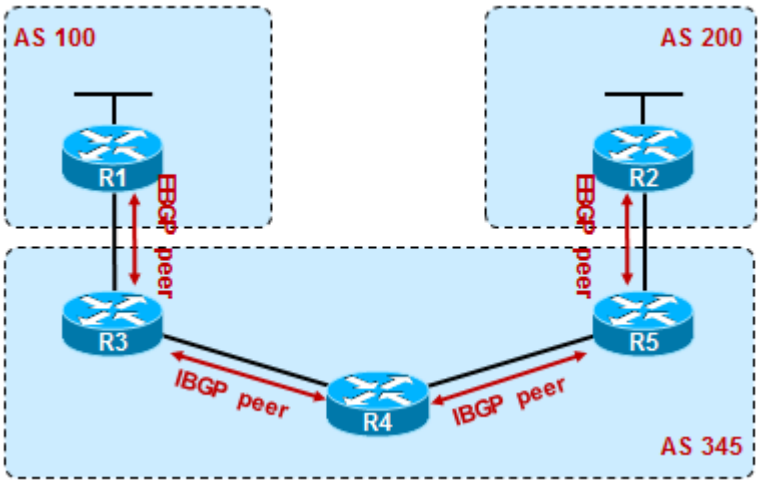
Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	0	100	0	200 i
*>i	3.3.3.3	0	100	0	100 i
* i100.0.2.0/24	5.5.5.5	0	100	0	200 i
*>i	3.3.3.3	0	100	0	100 i

发现 R4 上，前往 100.0.1.0 及 2.0 也存在冗余链路，分别通过 R3 和 R5 都能到达这两个网段，而最终 R4 选择 R3 作为前往这两个网段的下一跳，这是因为在 BGP 选路原则中，路由优选决策比较到了 BGP routerID 这一项，而 R3 的 RouterID 比 R5 要小，因此 R4 优选 R3。

到目前为止，我们已经完成了基本的配置，那么从 R4 上能 ping 同 100.0.1.1 及 2.1 么？试过之后我们一定得出否定的答案，这是因为在 R1 及 R2 上没有 AS345 内的路由，也就是回程路由出了问题，在这个拓扑中，我们可以在 R1 及 R2 上配置回程的路由或默认路由来完成实验的测试（注意，实际环境中，可能情况大不一样）。

实验完成到此处，我们设置 **MARK** 一下，接下去我们将分析各种属性对选路的影响。

7 通过修改 weight 来影响路由决策



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系，R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

WEIGHT 属性是 CISCO 私有属性。作用范围是本路由器(不传递),该值不既不会被包含在 update 消息中，也不会传递给任何 BGP 邻居。

范围 0-65535。

如果路由是从其他邻居学过来的，则默认值（在本地该路由）是 0

本地 network 产生的路由 weight 是 32768

本地重发布的直连接口路由、静态路由的 weight 为 32768

本地汇总产生的 BGP 路由 weight 值也为 32768

越大越优先

现在我们通过调整 weight 值，来影响 R4 上关于 100.0.1.0 及 2.0 路由的选择，使得 R4 优选 R5 作为去往 100.0.1.0 及 2.0 的下一跳。

在 R4 上

```
router bgp 345
  neighbor 5.5.5.5 weight 100
```

再观察一下 R4 的 BGP 表，我们发现：

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i100.0.1.0/24	5.5.5.5	0	100	100	200 i
* i	3.3.3.3	0	100	0	100 i
*>i100.0.2.0/24	5.5.5.5	0	100	100	200 i
* i	3.3.3.3	0	100	0	100 i

R4 选择了 R5 作为前往 100 网段的下一跳，这是因为这两条路由，从 R5 走，weight 权重值为 100，R3 的权重值为默认的 0，自然是 R5 要优选，因此 R4 将路径切换到 R5。

注意，这种修改方法，将对 R5 发来的所有路由生效（所有路由的权重都会变成 100），并且 weight 值的设置只能在本地做，且无法传递给任何 BGP 邻居。

如果希望通过修改 weight，让 R4 去往 100.0.1.0 走 R3 去往 100.0.2.0 走 R5 呢？那么配置修改如下：

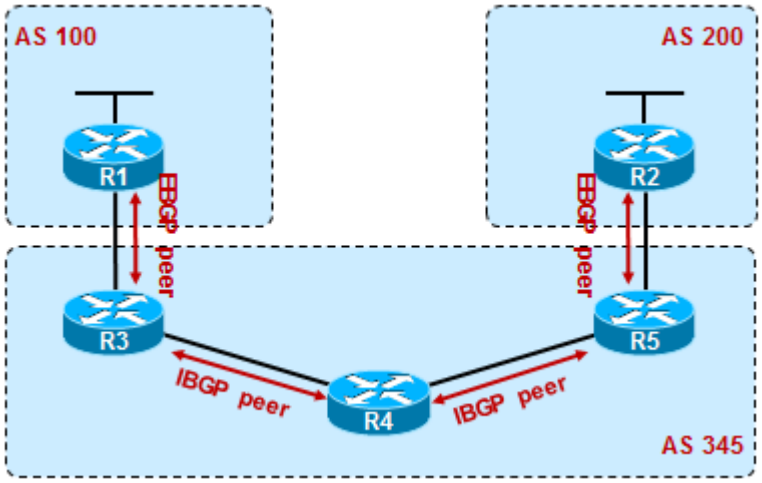
```
ip prefix-list 1 permit 100.0.1.0/24
```

```
ip prefix-list 2 permit 100.0.2.0/24
route-map WT2 permit 10
 match ip address prefix-list 1
 set weight 100
route-map WT2 permit 20
 match ip address prefix-list 2
 set weight 200
!
route-map WT1 permit 10
 match ip address prefix-list 1
 set weight 200
route-map WT1 permit 20
 match ip address prefix-list 2
 set weight 100

router bgp 345
neighbor 3.3.3.3 route-map WT1 in
neighbor 5.5.5.5 route-map WT2 in
```

配置完成后，使用 clear ip bgp * soft，软重置一下

8 通过修改 LOCAL_PREF 属性来影响流量



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系 ,R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

公认自决属性。LP 就是本地优先级，用于在内部对等体之间（IBGP）的 Update 消息，而不会传递给其他 EBGP 邻居，LP 值越大越优先。

1. 只能在 IBGP Peer 之间传递(除非做了策略否则 LP 值在 AS 内的 IBGP 邻居间传递不会丢失)，不能在 EBGP Peer 之间传递，如果在 EBGP Peer 之间收到的路由的路径属性中携带了 Local Preference ,则会触发 Notifacation 报文,造成会话中断 ;但是可以再 AS 边界路由器上使用 IN 方向的策略。
2. bgp default local-preference 500 //修改始发于本地的路由的默认 lp 值
3. BGP 路由器在向其 EBGP 邻居发送路由更新时，不能携带 LP 属性，对方收到该 EBGP 路由的 LP 值为空（连 LP 这个字段都木有），但是它会在本地为这条路由赋一个默认值，也就是 100，然后再传递给自己的 IBGP
4. 本地 network 及重发布的路由，LP 默认 100，并能在 AS 内向其他 IBGP 邻居传输，传输过程中除非部署策略，否则 LP 不变

我们可以通过设置 LP 值来影响路由，例如，同样实现让 R4 访问 100.0.1.0 走 R3，访问 100.0.2.0 走 R5。那么我们可以分别在 R3 及 R5 上对 R4 使用 OUT 方向的 route-map，并控制 LP 值

在 R3 使用策略

```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.2.0/24
route-map LP permit 10
  match ip address pref 1
  set local-preference 200
route-map LP permit 20
  match ip address pref 2
  set local-preference 100
router bgp 345
  neighbor 4.4.4.4 route-map LP out
```

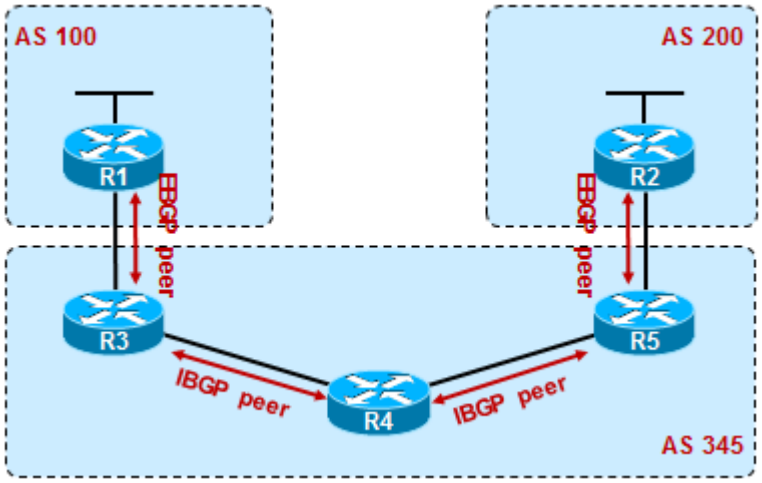
在 R5 上配置相似，只不过调整一下 LP 值

```
ip prefix-list 1 permit 100.0.1.0/24
ip prefix-list 2 permit 100.0.2.0/24
route-map LP permit 10
   match ip address pref 1
   set local-preference 100
route-map LP permit 20
   match ip address pref 2
   set local-preference 200
router bgp 345
   neighbor 4.4.4.4 route-map LP out
```

策略生效后查看 R4 的 BGP 表：

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	0	100	0	200 i
***>i**	**3.3.3.3**	**0**	**200**	**0**	**100 i**
***>i100.0.2.0/24**	**5.5.5.5**	**0**	**200**	**0**	**200 i**
* i	3.3.3.3	0	100	0	100 i

9 通过 AS_PATH 影响路由选择



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系 ,R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

AS_PATH 是公认必遵属性 ,描述到达目标网络所要经过的 AS 号序列。最重要的作用是防环，如果 BGP 发言者发现自己的 AS 号位于接收自外部对等体的路由，则忽略该路由

仅当 update 消息被发送给其他的 AS 时，BGP 路由器才会将其 AS 号追加在 AS_PATH 中。这句话也隐含了另一个意思，那就是如果要修改 AS_PATH 属性，则必须在 AS 边界路由器上执行策略。

R4 能学习到源自 R1 及 R2 的 100.0 网段路由，并且根据此前的实验结果，我们知道，在不部署任何策略的情况下，R4 优选 RouterID 小的 R3 作为去往 100.0 网段的 BGPpeer，那么我们能否通过控制 AS_PATH 来影响路由的优选呢？下面，我们使用 route-map 在 R1 上做策略，对 R3 生效，修改 100.0.1.0 路由的 AS_PATH 路径属性。最终使得 R4 去往 1.0 网络走 R2，去往 2.0 网络仍走 R1。

R1 的配置如下：

```
ip prefix-list 1 permit 100.0.1.0/24
route-map test permit 10
 match ip address prefix-list 1
 set as-path prepend 100
route-map test permit 20

router bgp 100
 no synchronization
 network 100.0.1.0 mask 255.255.255.0
 network 100.0.2.0 mask 255.255.255.0
 neighbor 10.1.13.3 remote-as 345
 neighbor 10.1.13.3 route-map test out
 no auto-summary
```

set as-path prepend 100 这条命令，会在路由的 AS_PATH 路径属性前插入 100 这个 AS 号，注意，要谨慎使用这种控制 AS_PATH 的方法，因为有可能导致隐患，这个实验只是通过拉长

AS_PATH 的长度，以影响路径选择，在实际的环境中不是特别建议这么操作。

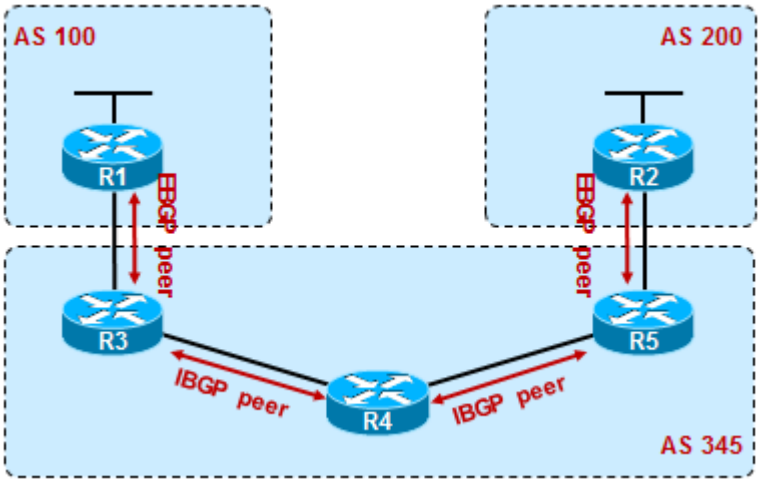
再来看一下 R4 的 BGP 表：

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i100.0.1.0/24	5.5.5.5	0	100	0	200 i
*** i**	**3.3.3.3**	**0**	**100**	**0**	**100 100 i**
*>i100.0.2.0/24	3.3.3.3	0	100	0	100 i
* i	5.5.5.5	0	100	0	200 i

我们发现 R4 上，由 R3 传递过来的 100.0.1.0 这条路由，AS_PATH 为 100 100，而从 R5 过来的是 200，明显从 R5 来的路由 AS_PATH 要短，因此优选经 R5 到 R2 的路由。

这个策略，我们也可在 R3 上做，只不过这时策略的执行方向是 in。

10 通过 original 属性影响路由选择



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系，R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

知识回顾

对于 original 属性，它明确了路由更新的来源，有以下几种取值：

标记	缩写	描述
IGP	i	通过 BGP 手工 network，也就是起源于 IGP，因为 BGP network 必须保证该网

		络在路由表中已有。
EGP	e	是由 EGP 这种早期的协议重发布而来
Incomplete	?	从其他渠道学习到的，路由来源不完全（确认该路由来源的信息不完全）。 (重发布 IGP 或静态)

并且优选顺序是：i > e > ?

为了验证这个规则，我们先保持基本的 BGP 配置。在 R1 上使用 network 的方式引入 100.0 的两条路由，在 R2 上我们使用**重发布的方式**引入此二条路由：

R1 的配置如下：

```
router bgp 100
network 100.0.1.0 mask 255.255.255.0
 network 100.0.2.0 mask 255.255.255.0
```

R2 的配置如下

```
ip prefix-list 1 seq 5 permit 100.0.1.0/24
ip prefix-list 2 seq 5 permit 100.0.2.0/24
route-map test permit 10
 match ip address prefix-list 1 2
router bgp 200
redistribute connected route-map test
```

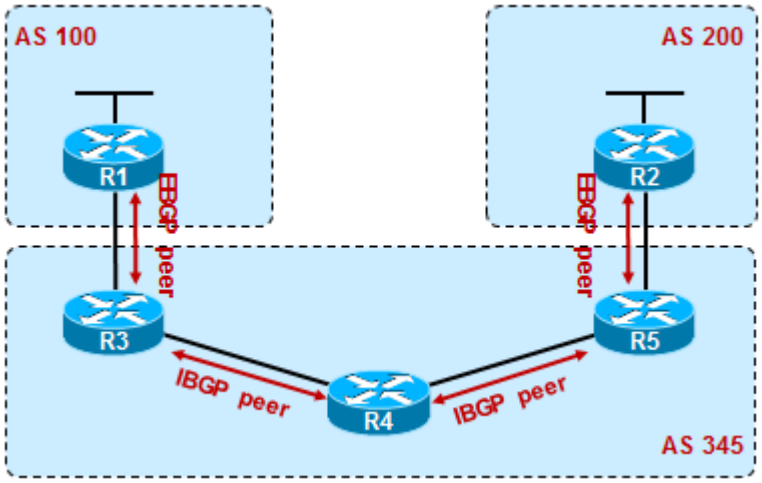
如此一来，在 R4 上，BGP 表如下：

```
R4#sh ip bgp
BGP table version is 4, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
                r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.1.0/24     5.5.5.5                  0    100      0 200 ?
*>i                 3.3.3.3                  0    100      0 100 i
*>i100.0.2.0/24     3.3.3.3                  0    100      0 100 i
* i                 5.5.5.5                  0    100      0 200 ?
```

我们发现，去往 100.0.1.0 及 2.0 的路由，下一跳均为 3.3.3.3，这是因为，这两条路径，original 值均为 “ i ”，要优于 “ ? ”

11 通过 MED 影响路由选择



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系，R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

CISCO 默认 MED 为 0，可选非传递

默认情况下，只比较来自同一邻居 AS 的 BGP 路由的 MED 值，就是说如果同一个目的地的两条路由来自不同的 AS，则不进行 MED 值的比较。MED 只是在直接相连的自治系统间影响业务量，而不会跨 AS 传递，MED 越小越优先。

我们仍然要实现让 R4 访问 100.0.1.0 网络走 R1，访问 100.0.2.0 网络走 R2，这一次我们通过调整 MED 实现，

R1 的配置如下：

```
ip prefix-list 1 seq 5 permit 100.0.1.0/24
ip prefix-list 2 seq 5 permit 100.0.2.0/24
route-map test permit 10
 match ip address prefix-list 1
```

```
 set metric 100
route-map test permit 20
 match ip address prefix-list 2
 set metric 200
router bgp 100
 network 100.0.1.0 mask 255.255.255.0
 network 100.0.2.0 mask 255.255.255.0
 neighbor 10.1.13.3 route-map test out
```

R2 的配置如下：

```
ip prefix-list 1 seq 5 permit 100.0.1.0/24
ip prefix-list 2 seq 5 permit 100.0.2.0/24
route-map test permit 10
 match ip address prefix-list 1
 set metric 200
route-map test permit 20
 match ip address prefix-list 2
 set metric 100
router bgp 100
 network 100.0.1.0 mask 255.255.255.0
 network 100.0.2.0 mask 255.255.255.0
 neighbor 10.1.25.5 route-map test out
```

上面的配置我们期望实现的效果是 R1 给 100.0.1.0 路由设置 MED=100 ,2.0 设置 MED=200，同时 R2 给 100.0.2.0 设置 MED=100，1.0 设置 MED 为 100，将这个属性分别对各自的 EBGP 邻居生效，于是乎给 MED 属性会随着路由在 AS345 内传递，最终抵达 R4，

那么按照我们的思路，在 R4 上应该能看到去往 100.0.1.0 选择的是 R3 也就是 3.3.3.3，去往 100.0.2.0 应该选择的是 R5，查看一下 R4 的 BGP 表：

```
R4#sh ip b
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	200	100	0	200 i
*>i	3.3.3.3	**100**	100	0	100 i
* i100.0.2.0/24	5.5.5.5	**100**	100	0	200 i

*>i	3.3.3.3	200	100	0	100 i

我们发现一个很奇怪的现象，R4 最终去往 100.0 的两个网络均选择了 3.3.3.3 作为下一跳，而不是按照 metric 这一属性来选路，为什么呢？原来 MED 属性，BGP 路由器默认只比较来源于同一宿主的路由的 MED 值，而 R1 及 R2 分属 AS100 及 AS200，那么对于 R4 来说,100.0.1.0 路由的两个路径来自不同的 AS，默认的它就绕过 MED 比较，而最终，影响选路的规则就是："BGP 邻居的 RID（越小越优先）"。

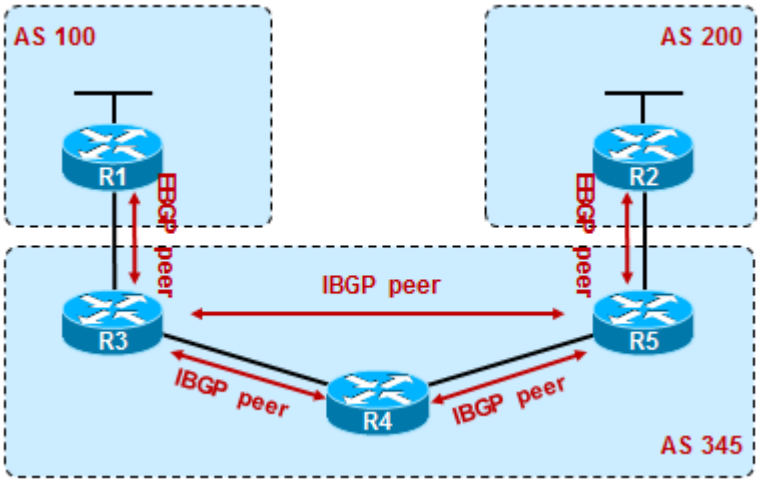
我们可以在 R4 上使用 bgp always-compare-med 命令，强制 R4 比较路由的 MED

最终：R4#sh ip b

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	200	100	0	200 i
*>i	3.3.3.3	100	100	0	100 i
*>i100.0.2.0/24	5.5.5.5	100	100	0	200 i
* i	3.3.3.3	200	100	0	100 i

选路就达到了我们的期望。

12 优选 EBGP 路由



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3、R4、R5 建立 IBGP 全互联，R3 R4 R5 使用 LOOPBACK 作为更新源并互指 neighbor。

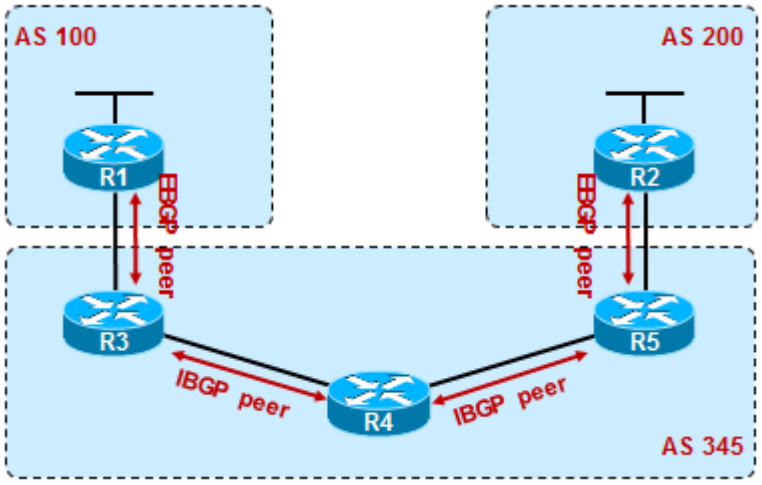
R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

优选 EBGP 邻居发来的路由（相对于 IBGP 邻居），在联邦 EBGP 和 IBGP 中优选联盟 EBGP 路由，注意，这条规则的生效，需保证 BGP 选路规则中的前面 6 条都无法做出决策的情况下才能达成。

完成上面描述的基本配置后，我们即可在 R5 上看到现象，R5 会同时从 R2 收到 100.0 网段的路由，也会从 R3 收到来自 R1 发布的 100 网段的路由，最终 R5 会优选 R2 的路由，这正匹配住了此条规则。

13 优选到 BGP NEXT_HOP 最近的路由



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3 及 R4、R4 及 R5 建立 IBGP 邻居关系，R3 R4 R5 使用 LOOPBACK 作为更新源并互指 neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

知识回顾

优选到 BGP NEXT_HOP 最近的路由，该路由是去往下一跳路由器 IGP 度量值最小的路由

在完成上述基本配置后，R4 的 BGP 表如下：

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	5.5.5.5	0	100	0	200 i
*>i	3.3.3.3	0	100	0	100 i
* i100.0.2.0/24	5.5.5.5	0	100	0	200 i

```
*>i                    3.3.3.3                  0    100      0    100 i
```

前面已经探讨过，我们在没有部署任何策略的情况下，该实验中 R4 之所以优选 3.3.3.3 也就是 R3 作为前往 100 网段的下一跳，原因是 R3 的 BGP RouterID 比 R5 的 RouterID 小，BGP 在进行决策规则比较时，一直比到了 RouterID 这一条，优选小的，因此选择了 R3。那么如何通过“NEXT_HOP”这一属性影响选路呢？

我们看看 100.0.1.0 这条路由：

R4#sh ip b 100.0.1.0

```
BGP routing table entry for 100.0.1.0/24, version 3
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  200
    5.5.5.5 (metric 2) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, internal
  100
    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

上述输出中，红色粗体的 metric 部分，就是本地到达 NEXT_HOP 属性地址的 IGP 度量值，我们发现从 R3 及 R5 走，metric 都是 2，那么如果我们将 R4 上连接 R3 的接口，也就是 F0/0 口的 OSPF cost 值调大，使得 R4 前往 3.3.3.3 路由的 OSPF metric 变大，会如何呢？

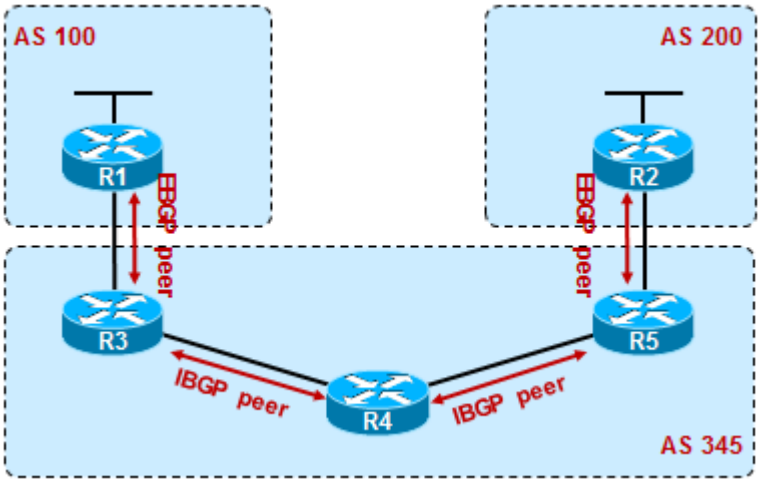
我们在 R4 的 F0/0 口，使用 ip ospf cost 100，将接口 cost 调大，于是结果如下：

R4#sh ip b 100.0.1.0

```
BGP routing table entry for 100.0.1.0/24, version 4
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  200
    5.5.5.5 (metric 2) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, internal, best
  100
    3.3.3.3 (metric 101) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

R4 上前往 3.3.3.3 的 OSPF metric 变成了 101，比前往 5.5.5.5 的 metric 要大，因此，R4 去往 100.0.1.0 的 BGP 路由，就优选了 R5。

14 使用路由反射器



我们将实验环境恢复为基本配置：BGP 的邻居关系如下

R1 及 R3、R2 及 R5 建立 EBGP 邻居关系；

R3及R4、R4及R5建立IBGP邻居关系，R3 R4 R5使用LOOPBACK作为更新源并互指neighbor。

R1 及 R2 都将 100.0.1.0 及 100.0.2.0 使用 network 的方式注入 BGP。

我们看一下 R5 的 BGP 表：

```
R5#sh ip bgp
   Network          Next Hop    Metric  LocPrf  Weight  Path
*> 100.0.1.0/24     10.1.25.2        0               0  200 i
*> 100.0.2.0/24     10.1.25.2        0               0  200 i
```

奇怪的现象：R5 的 BGP 表中，仅有 R2 通告的两条路由，而来自 R3 的 100.0 路由，却并未经由 R4 传递给 R5，换句话说，原本我们期望，AS345 去往 100.0 网络有冗余路径（通过 R1、R5 进行链路冗余），然而，此刻 R5 上却仅能通过 R2 去往目标网络。

造成这个现象的原因是？R4 并没将 R1 引入的、通过 R3 传递来的路由再传给 R5，

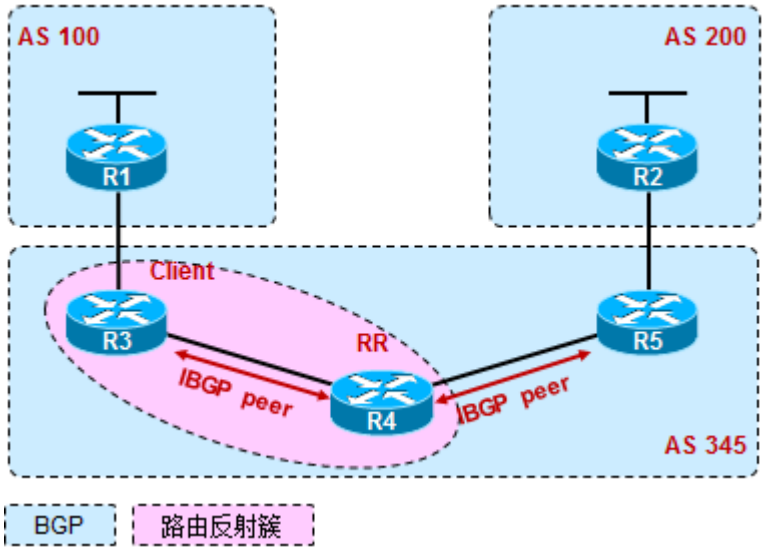
同时，R4 也收到了 R2 引入的 100.0 路由，但是并没有传递给 R3。这就是 IBGP 的水平分割原则：“BGP 路由器，不会将 IBGP 传递给他路由再传递给其他 IBGP 邻居”

那么解决的办法有哪些呢？

- 在传输 AS 内，建立 IBGP 全互联，也就是 R3 与 R4、R4 与 R5、R3 与 R5 之间都建立 IBGP 邻居关系

- 使用反射器
- BGP 联邦

建立 IBGP 全互联，配置比较简单，在上面的实验中已经有所涉及，我们来看看通过路由反射器如何实现。思路很简单，我们将 R4 配置为 RR，R3 为 R4 的 client，这样一来，R4 作为 RR，便会将学习自 Client R3 的路由，反射给 R5，也会将学习自 IBGP 邻居 R5 的路由，反射给 ClientR3。



R4 的 BGP 配置如下：

```
router bgp 345
 neighbor 3.3.3.3 remote-as 345
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 3.3.3.3 route-reflector-client
 neighbor 5.5.5.5 remote-as 345
 neighbor 5.5.5.5 update-source Loopback0
```

R4 将 R3 配置为 client，那么这就能突破水平分割的限制，将学习自 R3 的路由反射给 R5，将学习自 R5 的路由反射自 R3。再来看看 R5 的 BGP 表：

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	3.3.3.3	0	100	0	100 i
*>	10.1.25.2	0		0	200 i
* i100.0.2.0/24	3.3.3.3	0	100	0	100 i
*>	10.1.25.2	0		0	200 i

R5 上便看到了去往 100.0 网络的冗余路径，下一跳是 3.3.3.3，最终 R5 优选 R2 作为去往 100.0 网络，这是符合拓扑期望的，但是深层原因，还是由于相比于从 R3 来的 IBGP 路由，学习自 R2

的 EBGP 路由要更优。

看过了 R5 再去看看 R3：

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 100.0.1.0/24	10.1.13.1	0		0	100 i
*> 100.0.2.0/24	10.1.13.1	0		0	100 i

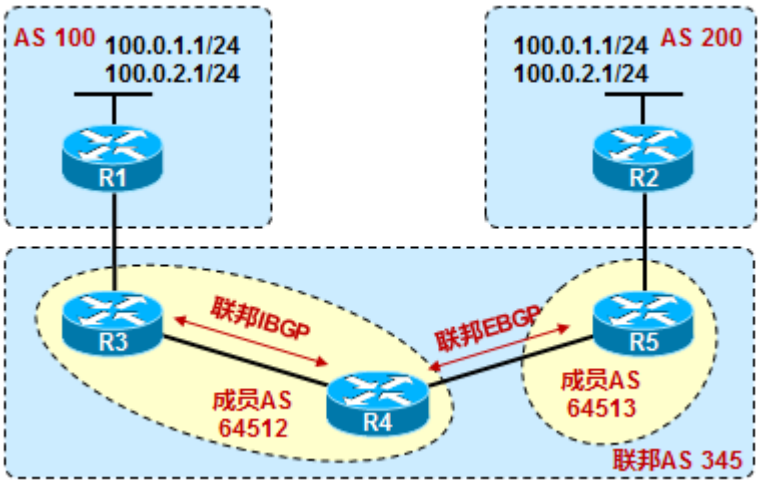
奇怪的现象出现了，似乎 R4 并没有把来自 R5 的路由反射给 R3，这是为什么？理论上说，路由反射器会将学习自非 client 的 IBGP 邻居发来的路由，反射给自己的 Client，但是显然，在这个实验中，R4 并没有将学习自 R5 的路由反射给自己的 Client。

我们还是上 R4 看看：

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i100.0.1.0/24	3.3.3.3	0	100	0	100 i
* i	5.5.5.5	0	100	0	200 i
*>i100.0.2.0/24	3.3.3.3	0	100	0	100 i
* i	5.5.5.5	0	100	0	200 i

发现什么了么？R4 的 BGP 表中，关于 100.0 网络都存在冗余路径，但是，最终经过 BGP 选路的决策，选了 R3 作为下一跳。而 BGP 仅将最优的路由（best，也就是>标记的）传递给 BGP 邻居，所以，R4 自然是不会把学习自 R3 的路由再发回给 R3 的。最后，虽然 R3 上 100.0 的两个网段都只存在一条路径，但是却不妨碍网络实现冗余性，当 R1 DOWN 掉后，R4 上便只有来自 R5 的路由，自然最后的也就是来自 R5 的路由，反射给 R3 也就没有问题了。

15 配置 BGP 联邦



联邦是解决 IBGP 全互联的另一种办法。我们可以通过在 AS 内（联邦 AS），划分一系列的小 AS，这些小 AS 被称作联邦成员 AS，在联邦成员内，BGP 路由器之间维护 IBGP 关系，遵循水平分割原则，在成员联邦 AS 之间，维护的是联邦的 EBGP 关系，不受 IBGP 水平分割的限制。而所有的联邦成员 AS，又都隶属于联邦 AS，对外统一使用联邦 AS 的标识。

在这个实验中，我们将 AS345 定义为联邦 AS，R3、R4 划分为成员 AS，使用 64512 作为 AS 号，R5 单独做为一个成员 AS，使用 64513 作为 AS 号。

R3 的配置如下：

```
router ospf 100
 network 3.3.3.3 0.0.0.0 area 0
 network 10.1.34.0 0.0.0.255 area 0
router bgp 64512
 no synchronization
 no auto-summary
 bgp confederation identifier 345
 neighbor 4.4.4.4 remote-as 64512
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 10.1.13.1 remote-as 100
```

注意，R3 创建的 BGP，使用的 AS 号是 64512，这是联邦成员 AS 号而不是联邦 AS 号 345；R3 有一个联邦 IBGP 邻居 R4。

bgp confederation identifier 345　　这条命令，用来告诉联邦外的 AS，我本地的 AS 号为 345

R4 的配置如下：

```
router bgp 64512
 no synchronization
 bgp confederation identifier 345
 bgp confederation peers 64513
 neighbor 3.3.3.3 remote-as 64512
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 5.5.5.5 remote-as 64513
 neighbor 5.5.5.5 ebgp-multihop 3
 neighbor 5.5.5.5 update-source Loopback0
 no auto-summary
```

R4 有两个 BGP 邻居，其中 R3 为其联邦 IBGP 邻居，R5 为其联邦 EBGP 邻居

为了让 R4 知道，R5 为其联邦的 EBGP 邻居，而不是普通的 EBGP 邻居，需要增加如下配置：

bgp confederation peers 64513

同时要注意，由于 R4 与 R5 为联邦的 EBGP 邻居关系，因此同样存在 TTL 为 1 的问题，如果二者使用 loopback 接口建立 BGP 关系，那么还需使用到 neighbor 5.5.5.5 ebgp-multihop 3 命令。

R5 的配置如下：

```
router bgp 64513
 no synchronization
 bgp confederation identifier 345
 bgp confederation peers 64512
 neighbor 4.4.4.4 remote-as 64512
 neighbor 4.4.4.4 ebgp-multihop 4
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 10.1.25.2 remote-as 200
 no auto-summary
```

如此一来路由传递就没有问题了。

在 R1 及 R2 上 network 100.0 的两个网络进 BGP，那么在 R4 上的 BGP 表：

Network	Next Hop	Metric	LocPrf	Weight	Path
* i100.0.1.0/24	10.1.13.1	0	100	0	100 i
*	10.1.25.2	0	100	0	(64513) 200 i
* i100.0.2.0/24	10.1.13.1	0	100	0	100 i
*	10.1.25.2	0	100	0	(64513) 200 i

我们看到 R4 上，关于 100.0 的两个网络存在冗余路径，但是却都不是 best，自然 R4 也无法使用这些路由，为什么不是 best 呢？原因在于下一跳，nexthop 不可达。

注意在联邦外引入的路由，next-hop 属性在联邦内部传递是不会发生改变的，即使在不同的成员 AS 间传递也是如此，因此需让 R3 及 R5 对 R4 修改 next-hop 属性，设定为自己的更新源 IP，这个你已经知道怎么做了吧？