

HCIP-Datcom 分解实验 - BGP

臧家林制作



- BGP 实验 1 : BGP 基本配置
- BGP 实验 2 : BGP 认证
- BGP 实验 3 : BGP 路由
- BGP 实验 4 : BGP 协议 fake-as
- BGP 实验 5 : BGP 自动路由汇总
- BGP 实验 6 : BGP 手动路由汇总
- BGP 实验 7 : BGP 路由反射器
- BGP 实验 8 : BGP 选路—Preferred Value
- BGP 实验 9 : BGP 选路—Local Preference
- BGP 实验 10 : BGP 选路—Next Hop
- BGP 实验 11 : BGP 选路—AS-Path
- BGP 实验 12 : BGP 选路—MED
- BGP 实验 13 : BGP 选路—Community
- BGP 实验 14 : BGP 路由黑洞
- BGP 实验 15 : BGP 联盟
- BGP 实验 16 : BGP 路由过滤
- BGP 实验 17 : BGP 路由引入
- BGP 实验 18 : BGP 缺省路由

=====

BGP 实验 1 : BGP 基本配置

早期发布的 BGP 的 3 个版本目前已经停止使用，当前使用的版本是 BGP-4，已经广泛应用于 ISP (Internet Service Provider) 之间。

BGP 虽然是一种动态路由协议，但它实际上本身并不产生路由、不发现路由、不计算路由，其主要功能是完成最佳路由的选择并在 BGP 邻居之间进行最佳路由的传递。BGP 选择了 TCP 作为其传输协议，端口号为 179.

BGP 支持无类域间路由 CIDR (Classless Inter-Domain Routing) ,并且采用了触发增量更新方式，这大大地减少了 BGP 在传播路由信息时占用的带宽，特别适用于在互联网上传播大量的路由信息。

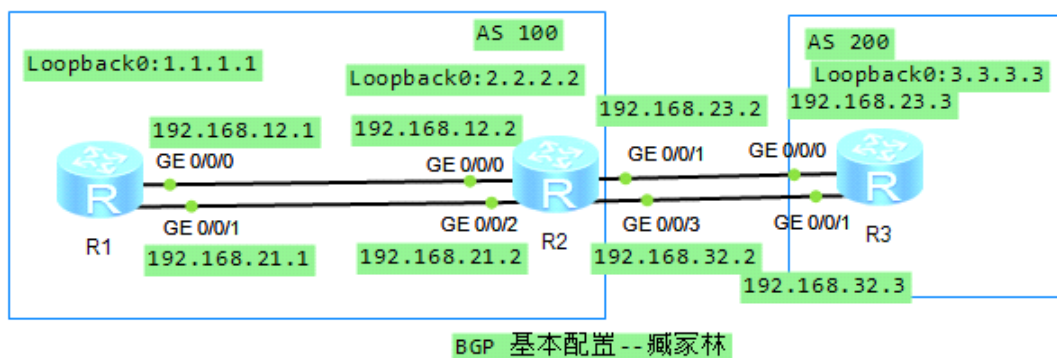
BGP 提供了丰富的路由属性 (Attribute)，通过对这些属性的操作和控制，BGP 能够非常容易地实现丰富而灵活的路由策略。BGP 还具有良好的扩展性，支持 Multicast，VPN，IPv6 等多种特性。

BGP 的邻居关系分为 IBGP (Internal BGP) 和 EBGP (External BGP) 两种：

当两台 BGP 路由器位于同一 AS 时 (AS 编号相同)，它们的邻居关系为 IBGP 邻居关系。

当两台 BGP 路由器位于不同的 AS 时 (AS 编号不同)，它们的邻居关系为 EBGP 邻居关系。

BGP 没有自动建立邻居关系的能力，邻居关系必须通过手动配置来建立。可以通过物理接口，或环回口来建立邻居关系。



基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.21.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
```

```
ip add 192.168.12.2 24
int g0/0/2
ip add 192.168.21.2 24
int g0/0/1
ip add 192.168.23.2 24
int g0/0/3
ip add 192.168.32.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/0
ip add 192.168.23.3 24
int g0/0/1
ip add 192.168.32.3 24
q
```

配置 IBGP 邻居

使用物理接口建立邻居

```
R1:
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 100
peer 192.168.21.2 as-number 100
q
```

R2:

```

bgp 100
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 192.168.21.1 as-number 100

```

[R2]display bgp peer

```
[R2]dis bgp peer
```

```

BGP local router ID : 2.2.2.2
Local AS number : 100
Total number of peers : 5                      Peers in established state : 2

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
192.168.12.1	4	100	14	13	0	00:11:01	Established	1
192.168.21.1	4	100	14	13	0	00:11:01	Established	1

在 R1 上将环回口 0 通告到 BGP 进程中

R1:

```

bgp 100
network 1.1.1.0 24

```

[R2]display bgp routing-table

```
[R2]dis bgp routing-table
```

```

BGP Local router ID is 2.2.2.2
Status codes: * - valid, > - best, d - damped,
               h - history, i - internal, s - suppressed, S - stale
Origin : i - IGP, e - EGP, ? - incomplete

```

```

Total Number of Routes: 2

```

Network	NextHop	MED	LocPrf	PrefVal	Path
*>i 1.1.1.0/24	192.168.12.1	0	100	0	i
* i	192.168.21.1	0	100	0	i

ignore 的应用

需要短暂中断邻居会话且该邻居配置量较大时，通过执行命令 peer ignore 可以避免重新配置的工作量。例如，在一段时间内，对端升级或调整链路导致邻居频繁建立连接时，为了避免路由或邻居关系频繁震荡，需要暂时中断 BGP 邻居，则可以

在较稳定的一端使用该命令。

使用该命令可以停止与指定对等体（组）之间的会话，并且清除所有相关路由信息。对于一个对等体组，这就意味着大量与对端的会话突然终止。

R1:

```
bgp 100
```

```
peer 192.168.12.2 ignore
```

邻居关系会变成 idle

Peer	V	AS	MsgRcvd	MsgSent
OutQ	Up/Down	State	PrefRcv	
192.168.12.2		4	100	0
0	0 00:00:03	Idle(Admin)		

=====

使用环回口来建立 BGP 邻居

使用 Loopback 接口建立 BGP 邻居关系比使用物理接口建立相比较，前者具有更好的稳定性，且能够减少设备资源的开销。

R1:

```
ip route-static 2.2.2.0 255.255.255.0 192.168.12.2
```

```
ip route-static 2.2.2.0 255.255.255.0 192.168.21.2
```

```
bgp 100
```

```
undo peer 192.168.12.2
```

```
undo peer 192.168.21.2
```

```
peer 2.2.2.2 as-number 100
```

peer 2.2.2.2 connect-interface LoopBack 0

R2:

ip route-static 1.1.1.0 255.255.255.0 192.168.12.1

ip route-static 1.1.1.0 255.255.255.0 192.168.21.1

bgp 100

undo peer 192.168.12.1

undo peer 192.168.21.1

peer 1.1.1.1 as-number 100

peer 1.1.1.1 connect-interface LoopBack 0

[R2]display bgp peer 邻居也能正常建立

[R2-bgp]dis bgp peer

BGP local router ID : 2.2.2.2

Local AS number : 100

Total number of peers : 1

Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
1.1.1.1	4	100	2	2	0	00:00:13	Established	0

配置 EBGP 邻居

使用 Loopback 0 接口来建立 EBGP 邻居

R2:

ip route-static 3.3.3.0 255.255.255.0 192.168.23.3

ip route-static 3.3.3.0 255.255.255.0 192.168.32.3

bgp 100

peer 3.3.3.3 as-number 200

peer 3.3.3.3 connect-interface LoopBack 0

peer 3.3.3.3 ebgp-max-hop

R3:

ip route-static 2.2.2.0 255.255.255.0 192.168.23.2

ip route-static 2.2.2.0 255.255.255.0 192.168.32.2

bgp 200

router-id 3.3.3.3

```
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface LoopBack 0
peer 2.2.2.2 ebgp-max-hop 255
```

<R2>display bgp peer 邻居建立成功

```
[R2]dis bgp peer
```

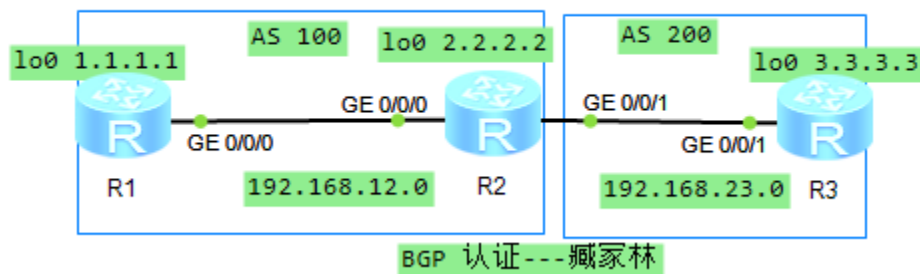
```
BGP local router ID : 2.2.2.2
Local AS number : 100
Total number of peers : 2                Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
1.1.1.1	4	100	5	5	0	00:03:02	Established
3.3.3.3	4	200	2	2	0	00:00:01	Established

通常使用 LoopBack 接口建立 IBGP 邻居，使用物理接口建立 EBGP 邻居

= = = = =

BGP 实验 2 : BGP 认证



物理接口 IP 地址来建立 BGP 邻居关系

认证是指路由器对路由信息来源的可靠性及路由信息本身的完整性进行检测的机制。

基本配置、配置 BGP 路由协议

R1:

```
undo ter mo
```

```
sys
```

```
sysname R1
```

```
int loop 0
```

```
ip add 1.1.1.1 24
```

```
int g0/0/0
```

```
ip add 192.168.12.1 24
```

```
q
```

```
bgp 100
```

```
router-id 1.1.1.1
```

```
peer 192.168.12.2 as-number 100
```

```
q
```

R2:

```
undo ter mo
```

```
sys
```

```
sysname R2
```

```
int loop 0
```

```
ip add 2.2.2.2 24
```

```
int g0/0/0
```

```
ip add 192.168.12.2 24
```

```
int g0/0/1
```

```
ip add 192.168.23.2 24
```

```
q
```

```
bgp 100
```

```
router-id 2.2.2.2
```

```
peer 192.168.12.1 as-number 100
```

```
peer 192.168.23.3 as-number 200
```

```
q
```

```

R3:
undo ter mo
sys
sysname R3
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.23.3 24
q

bgp 200
router-id 3.3.3.3
peer 192.168.23.2 as-number 100
q
配置完成后 BGP 邻居建立正常
<R2>display bgp peer

```

```
[R2]dis bgp peer
```

```

BGP local router ID : 2.2.2.2
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.1	4	100	2	2	0	00:00:18	Established
192.168.23.3	4	200	2	3	0	00:00:01	Established

=====

配置基于单一密钥的 BGP 认证功能

```

R1:
bgp 100
peer 192.168.12.2 password simple huawei

```

正常的 keepalive 60 hold 180

时间太长，认证之后，效果不明显，可以将计时器改的小一些

修改计时器

```
bgp 100
```

```
timer keepalive 5 hold 15
```

如果两端的时间不一致，会协商，最后选择小的时间，在配置时如果 hold 时间少于 20，会有提示信息的，提示邻居可能会有波动

```
dis bgp peer verbose
```

```
The hold time is less than 20 seconds
```

```
Configured: Active Hold Time: 180 sec
```

```
Keepalive Time:60 sec
```

```
Received      : Active Hold Time: 180 sec
```

```
Negotiated: Active Hold Time: 180 sec
```

```
Keepalive Time:60 sec
```

R1 做认证，R2 没做，邻居不能正常建立

```
[R1]dis bgp peer
```

```
BGP local router ID : 1.1.1.1
```

```
Local AS number : 100
```

```
Total number of peers : 1
```

```
Peers in established state : 0
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.2	4	100	0	0	0	00:00:32	Connect

R2:

```
bgp 100
```

```
peer 192.168.12.1 password simple huawei
```

```
peer 192.168.23.3 password cipher huawei
```

R3:

bgp 200

peer 192.168.23.2 password simple huawei

cipher 密钥会被加密，simple 是明文显示，但不影响邻居的建立

bgp 100

router-id 2.2.2.2

peer 192.168.12.1 as-number 100

peer 192.168.12.1 password simple huawei

peer 192.168.23.3 as-number 200

peer 192.168.23.3 password cipher ~:2YQKS[%;ECB7Ie7' /)ND\$#

#

=====

配置基于 Keychain 的 BGP 认证

选用 Periodic Daily 模式，每天 8 点到 18 使用 key-id 1 对 BGP 报文做 hash 运算。

先删掉 R1 R2 上的简单认证

R1 :

bgp 100

undo peer 192.168.12.2 password

R2 :

bgp 100

undo peer 192.168.12.1 password

接着做 key chain 方式的认证

```
[R1]keychain key mode ?  
  absolute Absolute timing mode  
  periodic  Periodic timing mode
```

absolute 绝对时间 periodic 周期时间

```
R1 :  
keychain key mode periodic daily  
key-id 1  
algorithm md5  
key-string huawei  
send-time daily 08:00 to 18:00  
receive-time daily 08:00 to 18:00  
bgp 100  
peer 192.168.12.2 keychain key
```

```
R2:  
keychain key mode periodic daily  
key-id 1  
algorithm md5  
key-string huawei  
send-time daily 08:00 to 18:00  
receive-time daily 08:00 to 18:00  
bgp 100  
peer 192.168.12.1 keychain key
```

```
[R1]display bgp peer 邻居关系是正常的
```

```
[R1]display keychain key
```

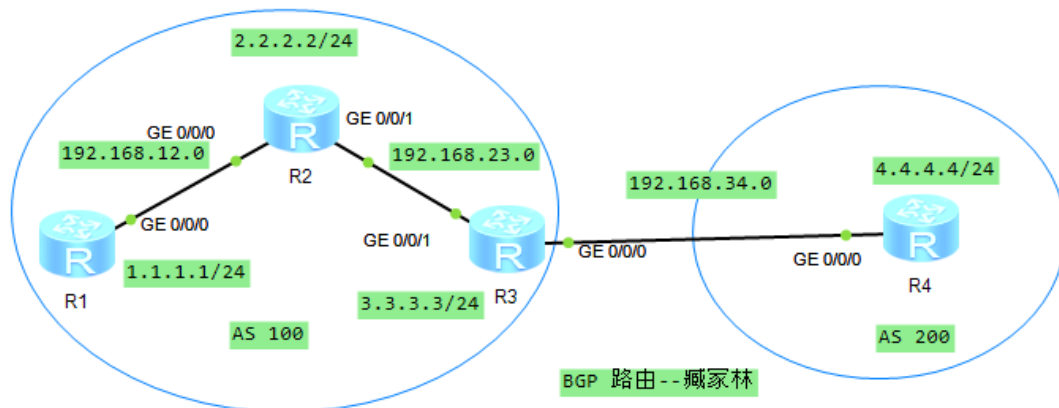
```
[R1-keychain]dis keychain key
```

Keychain Information:

```
-----  
Keychain Name           : key  
Timer Mode              : Daily periodic  
-----
```

=====

BGP 实验 3 : BGP 路由



IBGP 利用环回口建立邻居，IGP 协议为 OSPF，EBGP 通过物理接口建立邻居

IP 地址配置

```
R1:  
undo ter mo  
sys  
sysname R1  
user-interface console 0  
idle-timeout 0 0  
int loop 0  
ip add 1.1.1.1 24
```

```
int g0/0/0
ip add 192.168.12.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.23.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/0
ip add 192.168.34.3 24
int g0/0/1
ip add 192.168.23.3 24
q
```

```
R4:
undo ter mo
sys
```

```
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.34.4 24
q
```

= = = = =

配置 OSPF 协议

```
R1:
ospf router-id 1.1.1.1
area 0
net 1.1.1.1 0.0.0.0
net 192.168.12.1 0.0.0.0
q
```

```
R2:
ospf router-id 2.2.2.2
area 0
net 2.2.2.2 0.0.0.0
net 192.168.12.2 0.0.0.0
net 192.168.23.2 0.0.0.0
q
```

```
R3:
ospf router-id 3.3.3.3
area 0
net 3.3.3.3 0.0.0.0
net 192.168.23.3 0.0.0.0
q
```


在 BGP 100 内查看 OSPF 邻居关系，邻居是建立好的

```
[R2]dis ospf peer bri
```

OSPF Process 1 with Router ID 2.2.2.2 Peer Statistic Information			
Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	1.1.1.1	Full
0.0.0.0	GigabitEthernet0/0/1	3.3.3.3	Full

= = = = =

配置 BGP 协议

R1:

```
bgp 100
router-id 1.1.1.1
peer 2.2.2.2 as-n 100
peer 2.2.2.2 con loo0
peer 3.3.3.3 as-n 100
peer 3.3.3.3 con loo0
net 1.1.1.0 24
q
```

R2:

```
bgp 100
router-id 2.2.2.2
peer 1.1.1.1 as-n 100
peer 1.1.1.1 con loo0
peer 3.3.3.3 as-n 100
peer 3.3.3.3 con loo0
net 2.2.2.0 24
q
```

R3:

```
bgp 100
router-id 3.3.3.3
peer 2.2.2.2 as-n 100
```

```

peer 2.2.2.2 con loo0
peer 1.1.1.1 as-n 100
peer 1.1.1.1 con loo0
peer 192.168.34.4 as-n 200
net 3.3.3.0 24
q

```

```

R4:
bgp 200
router-id 4.4.4.4
peer 192.168.34.3 as-n 100
net 4.4.4.0 24
q

```

在 R3 上查看 BGP 的邻居关系，都是已建立

```

[R3]dis bgp peer
BGP local router ID : 3.3.3.3
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
1.1.1.1	4	100	5	7	0	00:02:33	Established
2.2.2.2	4	100	5	6	0	00:02:42	Established
192.168.34.4	4	200	5	8	0	00:02:34	Established

[R3]

查看 BGP 路由，在 R3 上学习到的 BGP 路由，都是可用且最优的

```

Total Number of Routes: 4

```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i 1.1.1.0/24	1.1.1.1	0	100	0	i
*>i 2.2.2.0/24	2.2.2.2	0	100	0	i
*> 3.3.3.0/24	0.0.0.0	0		0	i
*> 4.4.4.0/24	192.168.34.4	0		0	200i

[R3]

但在 R1 R2 上看到的路由是有问题的，

在 R1 上是无法到达 192.168.34.4 ，下一跳不可达，所以学习到的路由是不可用的
需要在 R3 进行改进

```
Total Number of Routes: 4
Network                NextHop          MED          LocPrf        PrefVal
*> 1.1.1.0/24           0.0.0.0          0             0             0
*>i 2.2.2.0/24           2.2.2.2          0             100           0
*>i 3.3.3.0/24           3.3.3.3          0             100           0
i 4.4.4.0/24           192.168.34.4     0             100           0
[R1]
```

R3 :
bgp 100
peer 1.1.1.1 next-hop-local
peer 2.2.2.2 next-hop-local

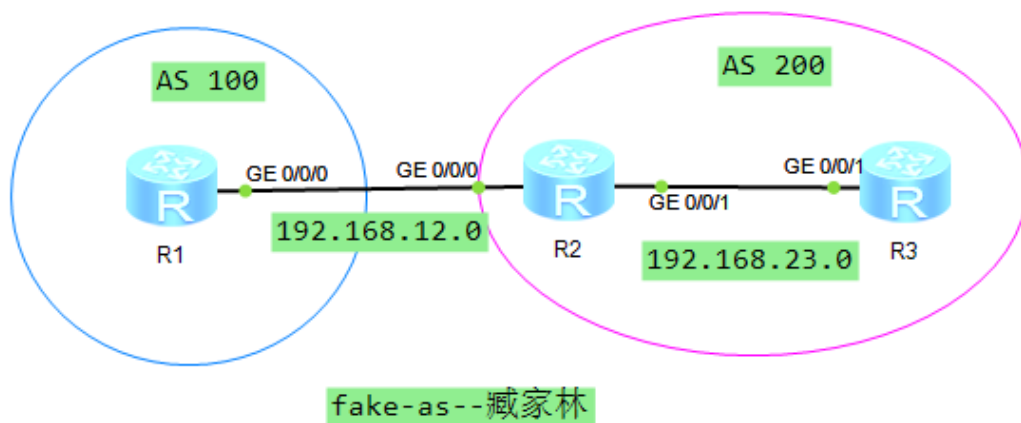
在路由传递在 R1 R2 的同时，把下一跳设置成自己，从而让路由变成可达
下一跳为 3.3.3.3 而 R1 是可以通过 OSPF 协议到在 3.3.3.3 的，路由可达了，就是可用且最优的

```
Total Number of Routes: 4
Network                NextHop          MED          LocPrf        PrefVal
*> 1.1.1.0/24           0.0.0.0          0             0             0
*>i 2.2.2.0/24           2.2.2.2          0             100           0
*>i 3.3.3.0/24           3.3.3.3          0             100           0
i 4.4.4.0/24           3.3.3.3          0             100           0
[R1]
```

=====

BGP 实验 4 : BGP 协议 fake-as

使用 fake-as 可以将本地真实的 AS 编号隐藏，其他 AS 内的对等体在指定本端对等体所在的 AS 编号时，应该设置成这个伪 AS 编号。



IBGP EBG 邻居都使用物理接口来建立

运行 BGP 的路由器一般情况下只能属于一个 AS，某些情况下，比如 AS 需要迁移或者和其他 AS 进行合并，要用新的 AS 替代原有的 AS 号码。

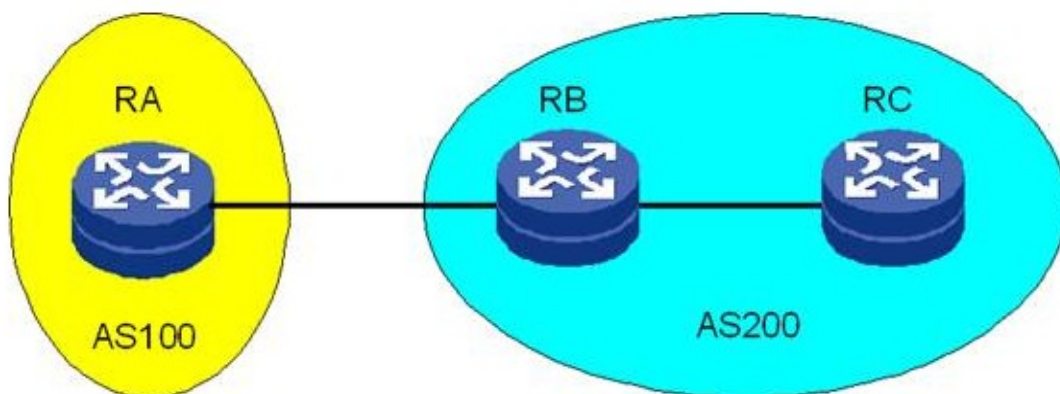


图 1 FAKE-AS 迁移前拓扑

迁移前网络拓扑如图 1 所示，RA 属于 AS100，RB 属于 AS200，相互之间建立 EBG 邻居。由于某些原因，用户要将 AS 200 替换成 AS2000，迁移后网络拓扑如图 2 所示：

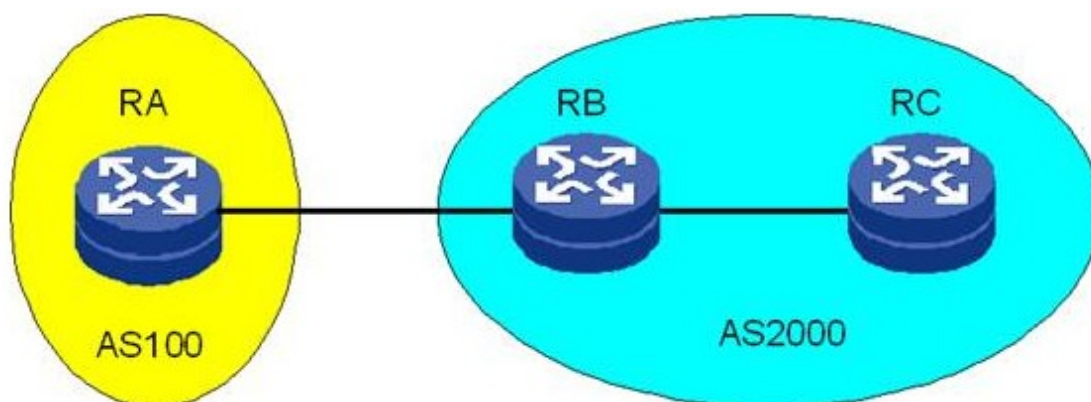


图2 FAKE-AS 迁移后拓扑

AS200 迁移为 AS2000 后，AS100 中的 RA 需要将邻居 RB 的 AS 号修改为 2000，但是由于某种原因 RA 的管理机构不愿意修改自己的配置，引入 FAKE-AS 就可以解决这个问题。

由于 RA 不修改自己的配置，所以至少需要解决以下 2 个问题：

1. BGP 在建立邻居的过程中会互相发送 OPEN 报文，要让 RA 认为 RB 发送过来的 OPEN 报文里面关于 AS 的记录是 200，而不是 2000；

2. RB 在发送给 RA 的路由，在 AS PATH 属性里面记录的第一个 AS 必须是 200，而不是 2000。

我们通过在 RB 上配置 FAKE-AS 功能可以完美的解决上述的 2 个问题，实现 RB 迁移 AS 的时候，RA 不需要修改任何 BGP 相关配置。

=====

基本 IP 地址配置

R1:
undo ter mo

```
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loo0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loo0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.23.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loo0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.23.3 24
```

q

= = = = =

配置 BGP 协议

R1:
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-n 200
q

R2:
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-n 100
peer 192.168.23.3 as-n 200
q

R3:
bgp 200
router-id 3.3.3.3
peer 192.168.23.2 as-n 200
q

正常配置好，邻居都能建立

[R2]dis bgp peer

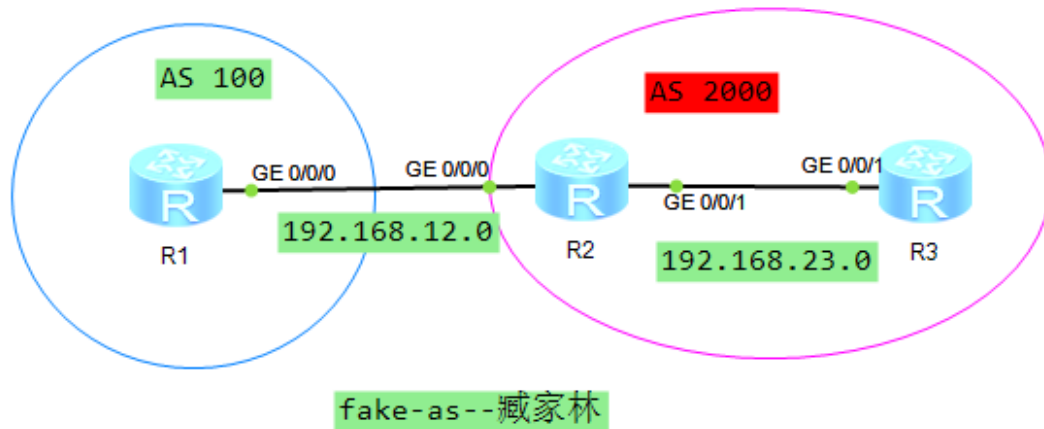
```
BGP local router ID : 2.2.2.2
Local AS number : 200
Total number of peers : 2
Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.1	4	100	4	4	0	00:02:52	Established
192.168.23.3	4	200	4	5	0	00:02:44	Established

[R2]

=====

配置 fake-as



AS 200 变成 AS 2000

R1 所在的 AS 100 不愿意改变配置指向 AS 2000 , 还是指向 AS 200

R2 R3 删掉 bgp 200 ,配置为 bgp 2000

R2:

undo bgp 200

R3:

undo bgp 200

=====

R2:

bgp 2000

router-id 2.2.2.2

peer 192.168.12.1 as-n 100

peer 192.168.23.3 as-n 2000

q


```

R3:
bgp 2000
router-id 3.3.3.3
peer 192.168.23.2 as-n 2000
q

```

改成 as 2000 后，R1 R2 之间是建立不了

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.1	4	100	0	0	0	00:00:09	Idle
192.168.23.3	4	2000	2	4	0	00:00:01	Established

[R2]

11	0.109000	192.168.12.2	192.168.12.1	TCP	63047	> bgp [FIN,
12	0.109000	192.168.12.2	192.168.12.1	BGP		OPEN Message
13	0.125000	192.168.12.1	192.168.12.2	TCP	bgp	> 63047 [ACK]
14	0.125000	192.168.12.1	192.168.12.2	BGP		NOTIFICATION Messa
15	0.125000	192.168.12.1	192.168.12.2	TCP	59977	> bgp [FIN,
16	0.140000	192.168.12.2	192.168.12.1	BGP		KEEPALIVE Message
17	0.140000	192.168.12.2	192.168.12.1	TCP	bgp	> 59977 [ACK]
18	0.140000	192.168.12.2	192.168.12.1	TCP	bgp	> 59977 [FIN,
19	0.156000	192.168.12.1	192.168.12.2	TCP	59977	> bgp [RST]
20	0.156000	192.168.12.1	192.168.12.2	TCP	59977	> bgp [RST]

Frame 12: 99 bytes on wire (792 bits), 99 bytes captured (792 bits)

Ethernet II, Src: HuaweiTe_70:36:bc (54:89:98:70:36:bc), Dst: HuaweiTe_18:68:16

Internet Protocol, Src: 192.168.12.2 (192.168.12.2), Dst: 192.168.12.1 (192.168.12.1)

Transmission Control Protocol, Src Port: bgp (179), Dst Port: 59977 (59977), Seq

Border Gateway Protocol

OPEN Message

Marker: 16 bytes

Length: 45 bytes

Type: OPEN Message (1)

Version: 4

My AS: 2000

Hold Time: 180

抓取 g0/0/0 的报文可以看到，R2 发给 R1 的 open 中自己的 as 为 2000

=====

R2 上配置 fake-as

```

R2:
bgp 2000
peer 192.168.12.1 fake-as 200

```

设置过 fake-as 之后，邻居就能建立了

```
[R2]dis bgp peer
```

```
BGP local router ID : 2.2.2.2
```

```
Local AS number : 2000
```

```
Total number of peers : 2
```

```
Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.1	4	100	2	2	0	00:00:01	Established
192.168.23.3	4	2000	3	5	0	00:01:40	Established
93 138.687000	192.168.12.2	192.168.12.1	TCP	57198 > bgp [ACK] S			
94 138.687000	192.168.12.2	192.168.12.1	BGP	OPEN Message			
95 138.703000	192.168.12.1	192.168.12.2	BGP	OPEN Message			
96 138.718000	192.168.12.2	192.168.12.1	BGP	KEEPALIVE Message			
97 138.734000	192.168.12.1	192.168.12.2	BGP	KEEPALIVE Message			
98 138.828000	192.168.12.2	192.168.12.1	TCP	57198 > bgp [ACK] S			

```
Frame 94: 99 bytes on wire (792 bits), 99 bytes captured (792 bits)
```

```
Ethernet II, Src: HuaweiTe_70:36:bc (54:89:98:70:36:bc), Dst: HuaweiTe_18:68:16
```

```
Internet Protocol, Src: 192.168.12.2 (192.168.12.2), Dst: 192.168.12.1 (192.168.
```

```
Transmission Control Protocol, Src Port: 57198 (57198), Dst Port: bgp (179), Seq
```

```
Border Gateway Protocol
```

```
□ OPEN Message
```

```
Marker: 16 bytes
```

```
Length: 45 bytes
```

```
Type: OPEN Message (1)
```

```
Version: 4
```

```
My AS: 200
```

```
Hold time: 180
```

```
BGP identifier: 2.2.2.2
```

配置之后，发送的 open 报文中自己的 as 为 200

== == == == == == == ==

BGP 实验 5：BGP 自动路由汇总

路由汇总，能有效地减少 BGP 路由器通告的路由条目数量，减小设备的路由表规模，并将拓扑变化产生的影响限制在一个相对更小的范围内。

自动路由聚合是在自然边界路由器上自动执行的，在默认情况下，BGP 的自动路由聚合功能是关闭的，并且 BGP 不会自动聚合 BGP 邻居发送的路由以及使用 network 命令通告的路由。

在大规模的网络中，BGP 路由表十分庞大，给设备造成了很大的负担，同时使发生路由振荡的几率也大大增加，影响网络

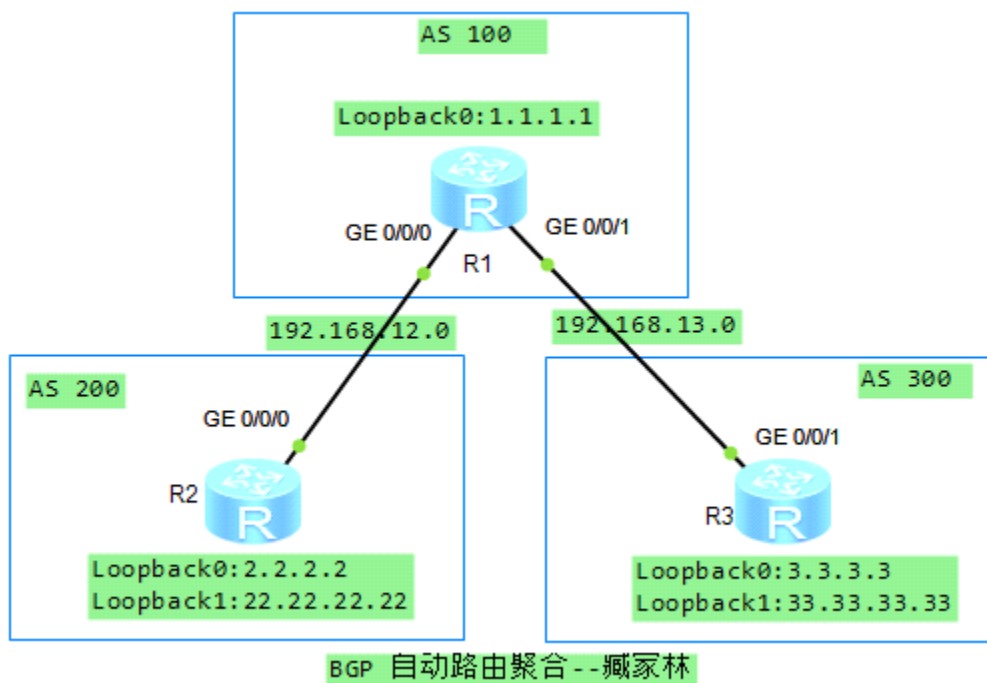
的稳定性。

路由聚合是将多条路由合并的机制，它通过只向对等体发送聚合后的路由而不发送所有的具体路由的方法，减小路由表的规模。并且被聚合的路由如果发生路由振荡，也不再对网络造成影响，从而提高了网络的稳定性。

BGP 在 IPv4 网络中支持自动聚合和手动聚合两种方式，而 IPv6 网络中仅支持手动聚合方式：

自动聚合：对 BGP 引入的路由进行聚合。配置自动聚合后，BGP 将按照自然网段聚合路由（例如非自然网段 A 类地址 10.1.1.1/24 和 10.2.1.1/24 将聚合为自然网段 A 类地址 10.0.0.0/8），并且 BGP 向对等体只发送聚合后的路由。

手动聚合：对 BGP 本地路由表中存在的路由进行聚合。手动聚合可以控制聚合路由的属性，以及决定是否发布具体路由。



使用物理接口建立 EBGP 邻居

基本配置

R1:

undo ter mo

sys

sysname R1

user-interface console 0

idle-timeout 0 0

int loop 0

ip add 1.1.1.1 24

int g0/0/0

ip add 192.168.12.1 24

int g0/0/1

ip add 192.168.13.1 24

q

R2:

undo ter mo

sys

sysname R2

user-interface console 0

idle-timeout 0 0

int loop 0

ip add 2.2.2.2 24

int loop 1

ip add 22.22.22.22 24

int g0/0/0

ip add 192.168.12.2 24

q

R3:

```
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int loop 1
ip add 33.33.33.33 24
int g0/0/1
ip add 192.168.13.3 24
q
```

配置 BGP 路由协议，使用物理接口建立邻居

```
R1 :
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
peer 192.168.13.3 as-number 300
q
```

```
R2 :
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
q
```

```
R3 :
bgp 300
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
q
```

[R1]display bgp peer 邻居能正常建立

```
BGP local router ID : 1.1.1.1
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
  192.168.12.2   4      200      2        3     0 00:00:08  Established
  192.168.13.3   4      300      2        3     0 00:00:01  Established
[R1]
```

=====

开启 BGP 自动路由聚合功能

默认是关闭的

R1:

bgp 100

summary automatic

R2:

bgp 200

summary automatic

R3:

bgp 300

summary automatic

当打开时，系统会有提示信息，说明 BGP 自动路由聚合只适用于通过路由引入方式引入的路由

```
[R1-bgp]summary automatic
Info: Automatic summarization is valid only for the routes imported through the
import-route command.
[R1-bgp]
```

通告路由进入 BGP 中

R3:

bgp 300

network 33.33.33.0 24

[R1]display bgp routing-table 查看 BGP 路由表

```
Total Number of Routes: 1
      Network                NextHop          MED           LocPrf        PrefVal
* >  33.33.33.0/24          192.168.13.3      0
[R1]
```

R3 的 33.33.33.0 没有被聚合，它是通过 network 通告的

引入外部路由到 BGP 协议中

R2:

bgp 200

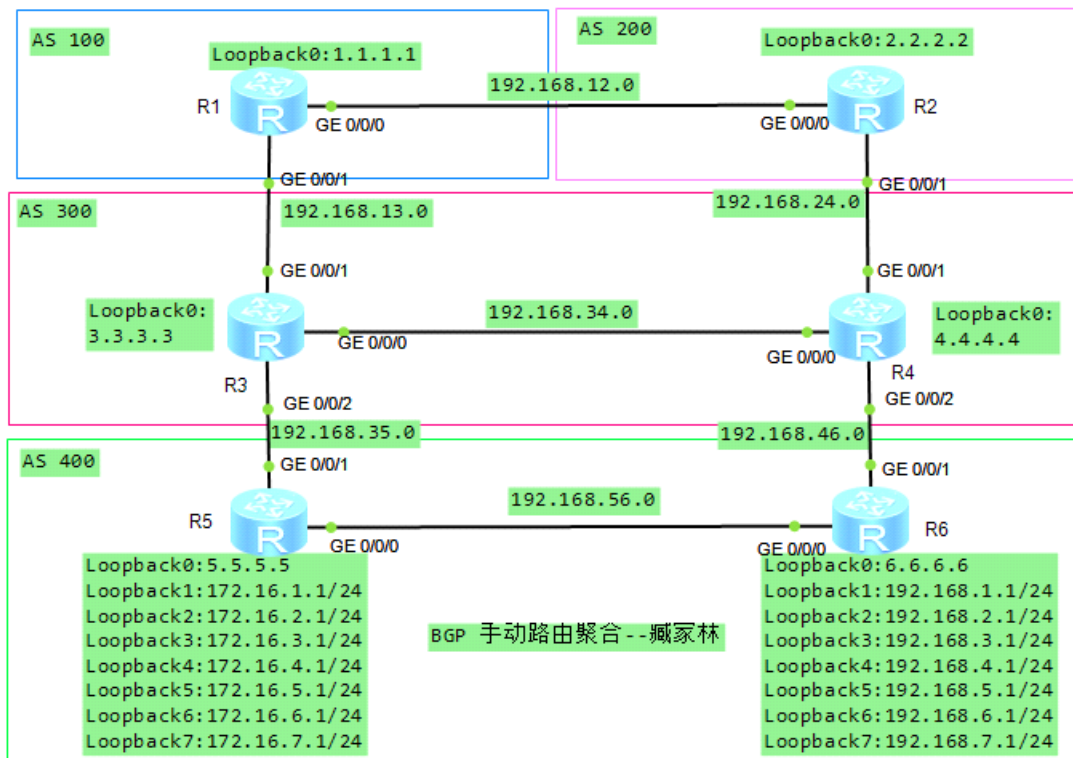
import-route direct

<R1>display bgp routing-table 可以看到 R1 的 BGP 表中，出现了 22.0.0.0 且没有显示掩码信息，这正是聚合后的一个 A 类自然网络。在实际项目中，很少启用 BGP 自动路由聚合功能。

```
Total Number of Routes: 4
      Network                NextHop          MED           LocPrf        PrefVal
* >  2.0.0.0                 192.168.12.2
* >  22.0.0.0                192.168.12.2
* >  33.33.33.0/24          192.168.13.3      0
      192.168.12.0          192.168.12.2      0
```

= = = = =

BGP 实验 6 : BGP 手动路由汇总



相对于自动路由聚合来讲，手动路由聚合具有更高的灵活性和可控性。

手动路由聚合时，可以手动控制聚合路由的掩码长度，修改聚合路由属性等。

物理接口建立 IBGP , EBGp 邻居

有两种方法：

- 1.配置一条静态路由，然后 network 命令进行通告
- 2.使用 aggregate 命令进行聚合。

基本配置

R1:


```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.13.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
```

```
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.13.3 24
int g0/0/0
ip add 192.168.34.3 24
int g0/0/2
ip add 192.168.35.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/1
ip add 192.168.24.4 24
int g0/0/0
ip add 192.168.34.4 24
int g0/0/2
ip add 192.168.46.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
```

```
int loop 1
ip add 172.16.1.1 24
int loop 2
ip add 172.16.2.1 24
int loop 3
ip add 172.16.3.1 24
int loop 4
ip add 172.16.4.1 24
int loop 5
ip add 172.16.5.1 24
int loop 6
ip add 172.16.6.1 24
int loop 7
ip add 172.16.7.1 24
int g0/0/0
ip add 192.168.56.5 24
int g0/0/1
ip add 192.168.35.5 24
q
```

```
R6:
undo ter mo
sys
sysname R6
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 6.6.6.6 24
int loop 1
ip add 192.168.1.1 24
int loop 2
ip add 192.168.2.1 24
int loop 3
ip add 192.168.3.1 24
```

```
int loop 4
ip add 192.168.4.1 24
int loop 5
ip add 192.168.5.1 24
int loop 6
ip add 192.168.6.1 24
int loop 7
ip add 192.168.7.1 24
int g0/0/0
ip add 192.168.56.6 24
int g0/0/1
ip add 192.168.46.6 24
q
```

配置 BGP 路由协议

使用直连物理接口建立邻居，R5 R6 的环回口在网络中使用 network 通告到 BGP 进程中

R1:

```
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
peer 192.168.13.3 as-number 300
q
```

R2:

```
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 192.168.24.4 as-number 300
q
```

R3:

```
bgp 300
router-id 3.3.3.3
```

```
peer 192.168.13.1 as-number 100
peer 192.168.34.4 as-number 300
peer 192.168.34.4 next-hop-local
peer 192.168.35.5 as-number 400
q
```

```
R4:
bgp 300
router-id 4.4.4.4
peer 192.168.24.2 as-number 200
peer 192.168.34.3 as-number 300
peer 192.168.34.3 next-hop-local
peer 192.168.46.6 as-number 400
q
```

```
R5:
bgp 400
router-id 5.5.5.5
peer 192.168.35.3 as-number 300
peer 192.168.56.6 as-number 400
peer 192.168.56.6 next-hop-local
network 172.16.1.0 24
network 172.16.2.0 24
network 172.16.3.0 24
network 172.16.4.0 24
network 172.16.5.0 24
network 172.16.6.0 24
network 172.16.7.0 24
q
```

```
R6:
bgp 400
router-id 6.6.6.6
peer 192.168.46.4 as-number 300
```

```

peer 192.168.56.5 as-number 400
peer 192.168.56.5 next-hop-local
network 192.168.1.0 24
network 192.168.2.0 24
network 192.168.3.0 24
network 192.168.4.0 24
network 192.168.5.0 24
network 192.168.6.0 24
network 192.168.7.0 24
q

```

配置完成后，查看相互间的邻居关系都要建立成功

<R3>display bgp peer

[R3]dis bgp peer

```

BGP local router ID : 3.3.3.3
Local AS number : 300
Total number of peers : 3                Peers in established state : 3

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.13.1	4	100	5	5	0	00:01:26	Established
192.168.34.4	4	300	3	6	0	00:01:11	Established
192.168.35.5	4	400	9	6	0	00:00:55	Established

[R3]

<R1>display bgp routing-table

R1 上接收到 BGP 协议的明细路由

```

Total Number of Routes: 28

```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 172.16.1.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.2.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.3.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.4.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.5.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.6.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 172.16.7.0/24	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 192.168.1.0	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i
*> 192.168.2.0	192.168.13.3			0	300 400i
*	192.168.12.2			0	200 300 400i

在 R5 上使用 ping 测试 R5 的环回口与 R6 的环回口之间的连通性

<R5>ping -a 172.16.1.1 192.168.1.1

```
[R5]ping -a 172.16.1.1 192.168.1.1
  PING 192.168.1.1: 56 data bytes, press CTRL_C to break
    Reply from 192.168.1.1: bytes=56 Sequence=1 ttl=255 time=30 ms
    Reply from 192.168.1.1: bytes=56 Sequence=2 ttl=255 time=30 ms
    Reply from 192.168.1.1: bytes=56 Sequence=3 ttl=255 time=30 ms
    Reply from 192.168.1.1: bytes=56 Sequence=4 ttl=255 time=30 ms
    Reply from 192.168.1.1: bytes=56 Sequence=5 ttl=255 time=60 ms
```

=====

配置 BGP 路由聚合

R6 上配置静态路由，然后使用 network 通告出去（第一种聚合方法）

```
R6:
ip route-static 192.168.0.0 21 NULL 0
bgp 400
network 192.168.0.0 21
q
```

静态路由指向 NULL0 的目的是防止网络中产生环路，这条静态路由仅仅是用来通告的

R5 上使用 aggregate（第二种聚合方法）

```
R5:
bgp 400
aggregate 172.16.0.0 255.255.248.0
q
```

可以写 21，也可以写 255.255.248.0，建议写成 255.255.248.0

使用 aggregate 命令进行聚合，要求 BGP 路由表中至少有一

条属于聚合后的路由的子网路由，否则聚合不会生效。

<R1>display bgp routing-table

```
Total Number of Routes: 32
Network                NextHop          MED           LocPrf        PrefVal Path/Ogn
*> 172.16.0.0/21        192.168.13.3    0             0             300 400i
* 172.16.0.0/21        192.168.12.2    0             0             200 300 400i
*> 172.16.1.0/24        192.168.13.3    0             0             300 400i
* 172.16.1.0/24        192.168.12.2    0             0             200 300 400i
*> 172.16.2.0/24        192.168.13.3    0             0             300 400i
* 172.16.2.0/24        192.168.12.2    0             0             200 300 400i

*> 192.168.0.0/21      192.168.13.3    0             0             300 400i
* 192.168.0.0/21      192.168.12.2    0             0             200 300 400i
*> 192.168.1.0         192.168.13.3    0             0             300 400i
* 192.168.1.0         192.168.12.2    0             0             200 300 400i
*> 192.168.2.0         192.168.13.3    0             0             300 400i
* 192.168.2.0         192.168.12.2    0             0             200 300 400i
```

R1 的 BGP 路由表包含了聚合后的路由，但同时也有明细路由。

=====

控制路由聚合，R5 R6 的方法有所不同

若想抑制明细路由，则需要配合 route-policy 来实现对明细路由的抑制。如果抑制明细路由的数量比较多时，配置工作量就会比较多，而且维护起来不太方便，这种方法的扩展性和可维护性比较差。

R6:

```
ip ip-prefix noad permit 192.168.1.0 24
ip ip-prefix noad permit 192.168.2.0 24
ip ip-prefix noad permit 192.168.3.0 24
ip ip-prefix noad permit 192.168.4.0 24
ip ip-prefix noad permit 192.168.5.0 24
ip ip-prefix noad permit 192.168.6.0 24
ip ip-prefix noad permit 192.168.7.0 24
ip ip-prefix noad permit 172.16.1.0 24
ip ip-prefix noad permit 172.16.2.0 24
```



```
ip ip-prefix noad permit 172.16.3.0 24
ip ip-prefix noad permit 172.16.4.0 24
ip ip-prefix noad permit 172.16.5.0 24
ip ip-prefix noad permit 172.16.6.0 24
ip ip-prefix noad permit 172.16.7.0 24
route-policy no-ad permit node 1
if-match ip-prefix noad
apply community no-advertise
route-policy no-ad permit node 2
bgp 400
peer 192.168.46.4 route-policy no-ad export
peer 192.168.46.4 advertise-community
```

使用前缀列表和 route-policy 给这些明细路由添加 no-advertise 的团体属性。

peer X.X.X.X advertise-community 是为了将团体属性传递给对等体，缺省情况下是不传递的。

R5 :

```
ip ip-prefix noad permit 192.168.1.0 24
ip ip-prefix noad permit 192.168.2.0 24
ip ip-prefix noad permit 192.168.3.0 24
ip ip-prefix noad permit 192.168.4.0 24
ip ip-prefix noad permit 192.168.5.0 24
ip ip-prefix noad permit 192.168.6.0 24
ip ip-prefix noad permit 192.168.7.0 24
ip ip-prefix noad permit 172.16.1.0 24
ip ip-prefix noad permit 172.16.2.0 24
ip ip-prefix noad permit 172.16.3.0 24
ip ip-prefix noad permit 172.16.4.0 24
ip ip-prefix noad permit 172.16.5.0 24
ip ip-prefix noad permit 172.16.6.0 24
ip ip-prefix noad permit 172.16.7.0 24
route-policy no-ad permit node 1
```

```

if-match ip-prefix noad
apply community no-advertise
route-policy no-ad permit node 2
bgp 400
peer 192.168.35.3 route-policy no-ad export
peer 192.168.35.3 advertise-community

```

配置完成后，在 R1 上查看一下，只有聚合路由，没有明细
<R1>display bgp routing-table

```

[R1]dis bgp routing-table
BGP Local router ID is 1.1.1.1
Status codes: * - valid, > - best, d - damped,
               h - history, i - internal, s - suppressed, S - stale
               Origin : i - IGP, e - EGP, ? - incomplete

Total Number of Routes: 4

```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 172.16.0.0/21	192.168.13.3			0	300 400i
*> 192.168.0.0/21	192.168.12.2			0	200 300 400i
*> 192.168.0.0/21	192.168.13.3			0	300 400i
*> 192.168.0.0/21	192.168.12.2			0	200 300 400i

<R3>display bgp routing-table R3 有聚合也有明细

```

Total Number of Routes: 18

```

Network	NextHop	MED	LocPrf
*> 172.16.0.0/21	192.168.35.5		
* i 172.16.0.0/21	192.168.34.4		100
*> 172.16.1.0/24	192.168.35.5	0	
*> 172.16.2.0/24	192.168.35.5	0	
*> 172.16.3.0/24	192.168.35.5	0	
*> 172.16.4.0/24	192.168.35.5	0	
*> 172.16.5.0/24	192.168.35.5	0	
*> 172.16.6.0/24	192.168.35.5	0	
*> 172.16.7.0/24	192.168.35.5	0	

=====

detail-suppressed 抑制明细
as-set 加上原有的 as-path

还可以利用 aggregate 命令结合 detail-suppressed 来实现对

明细路由的抑制，只将聚合后的路由发送出去。

R5 R6 都采用 aggregate 方式来聚合路由

R6:

```
undo ip route-static 192.168.0.0 21 NULL 0
```

```
bgp 400
```

```
undo network 192.168.0.0 21
```

```
aggregate 192.168.0.0 21 detail-suppressed
```

q

R5:

```
bgp 400
```

```
aggregate 172.16.0.0 21 detail-suppressed
```

R5 上只利用 aggregate 聚合 172.16.0.0 网段的，所以 detail-suppressed 抑制的是 172.16.0.0

<R3>display bgp routing-table

```
Total Number of Routes: 4
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/
*>	172.16.0.0/21	192.168.35.5			0	400i
* i		192.168.34.4		100	0	400i
*>	192.168.0.0/21	192.168.35.5			0	400i
* i		192.168.34.4		100	0	400i

[R3]

R5:

```
bgp 400
```

```
aggregate 172.16.0.0 21 detail-suppressed as-set
```

如果 172.16.0.0 来自多个 AS ,可以加上 as-set

除此之外，还有 suppress-policy 抑制部分明细，attribute-policy 可以修改聚合后的路由属性。

=====

suppress-policy 抑制部分明细

suppress-policy 抑制部分明细，抑制 172.16.2.0、172.16.4.0、172.16.6.0

R5:

```
ip ip-prefix sup permit 172.16.2.0 24
ip ip-prefix sup permit 172.16.4.0 24
ip ip-prefix sup permit 172.16.6.0 24
route-policy sup permit node 10
if-match ip-prefix sup
bgp 400
undo peer 192.168.35.3 route-policy noad export
aggregate 172.16.0.0 21 suppress-policy sup
```

配置完成后，在 R3 上查看一下

```
*> 172.16.0.0/21      192.168.35.5      0      400i
* i      192.168.34.4      100      0      400i
*> 172.16.1.0/24      192.168.35.5      0      0      400i
*> 172.16.3.0/24      192.168.35.5      0      0      400i
*> 172.16.5.0/24      192.168.35.5      0      0      400i
*> 172.16.7.0/24      192.168.35.5      0      0      400i
*> 192.168.0.0/21     192.168.35.5      0      0      400i
* i      192.168.34.4      100      0      400i
[R3]
```

=====

attribute-policy 可以修改聚合后的路由属性

R5 通告的是属性没有经过任何修改的路由，其聚合路由的 Origin 属性是 i。现在通过 attribute-policy 将聚合路由的属性修改成 Incomplete，在 BGP 路由表中显示为 ？

R5:

```

route-policy att permit node 10
apply origin incomplete
bgp 400
aggregate 172.16.0.0 21 attribute-policy att
suppress-policy sup

```

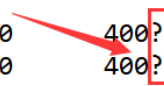
可以加 suppress-policy sup，也可以不加

配置完成后，在 R3 上查看一下

```

*> 172.16.0.0/21      192.168.35.5      0      400?
* i      192.168.34.4      100      0      400?
*> 172.16.1.0/24      192.168.35.5      0      0      400i
*> 172.16.3.0/24      192.168.35.5      0      0      400i
*> 172.16.5.0/24      192.168.35.5      0      0      400i
*> 172.16.7.0/24      192.168.35.5      0      0      400i
*> 192.168.0.0/21     192.168.35.5      0      0      400i
* i      192.168.34.4      100      0      400i
[R3]

```



=====

origin-policy

如果希望汇总路由的产生，只以某条或某些特定的明细路由为触发条件

如 R5 的 BGP 路由表中存在有 172.16.1.0/24 的路由时，才触发其产生汇总路由，172.16.0.0/21，而当明细路由 172.16.1.0/24 丢失，则 R5 不再通告该条汇总路由，那么就需要使用 origin-policy 关键字了。

R5：

```
ip ip-prefix or permit 172.16.1.0 24
```

```

route-policy or permit node 10
if-match ip-prefix or

```

```
bgp 400
```

aggregate 172.16.0.0 21 origin-policy or

配置完成后，R3 的路由中，是有汇总路由 172.16.0.0 /21 的

```
*> 172.16.0.0/21      192.168.35.5
* i                    192.168.34.4
*> 172.16.1.0/24      192.168.35.5      0
*> 172.16.2.0/24      192.168.35.5      0
```

在 R5 上把 int loo1 删除掉，没有了 172.16.1.0/24，在 R3 上也没有汇总路由了

R5：

undo int loo1

在 R3 上查看 BGP 路由表，可以看到没有 172.16.1.0，也没有汇总的路由

	Network	NextHop	MED
*>	172.16.2.0/24	192.168.35.5	0
*>	172.16.3.0/24	192.168.35.5	0
*>	172.16.4.0/24	192.168.35.5	0

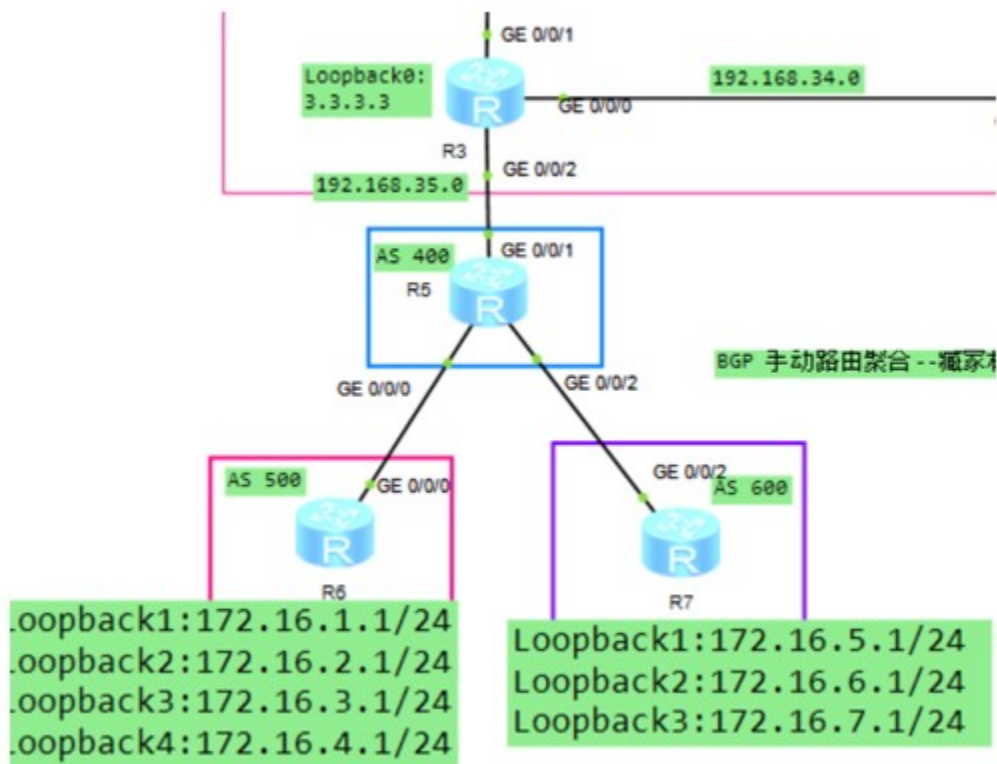
=====

在上面的实验基础上进行修改

as-set

为了避免路由聚合可能引起的路由环路，BGP 设计了 AS_Set 属性。AS_Set 属性是一种无序的 AS_Path 属性，标明聚合路

由所经过的 AS 号。当聚合路由重新进入 AS_Set 属性中列出的任何一个 AS 时，BGP 将会检测到自己的 AS 号在聚合路由的 AS_Set 属性中，于是会丢弃该聚合路由，从而避免了路由环路形成。



R5:

undo bgp 400 确实 Y

```
int g0/0/2
```

```
ip add 192.168.57.5 24
```

```
q
```

```
bgp 400
```

```
router-id 5.5.5.5
```

```
peer 192.168.56.6 as-n 500
```

```
peer 192.168.57.7 as-n 600
```

```
peer 192.168.35.3 as-n 300
q
```

```
R6 :
int loo1
ip add 172.16.1.1 24
int loo2
ip add 172.16.2.1 24
int loo3
ip add 172.16.3.1 24
int loo4
ip add 172.16.4.1 24
q
```

und bgp 400 , 要进行确认 Y

```
bgp 500
router-id 6.6.6.6
peer 192.168.56.5 as-n 400
net 172.16.1.0 24
net 172.16.2.0 24
net 172.16.3.0 24
net 172.16.4.0 24
q
```

```
R7:
undo ter mo
sys
sysname R7
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 7.7.7.7 24
int loo1
```



```
ip add 172.16.5.1 24
int loo2
ip add 172.16.6.1 24
int loo3
ip add 172.16.7.1 24
int g0/0/0
ip add 192.168.57.7 24
q
bgp 600
router-id 7.7.7.7
peer 192.168.57.5 as-n 400
net 172.16.5.0 24
net 172.16.6.0 24
net 172.16.7.0 24
q
```

在没有汇总之前，R3 看到的是明细路由
R3: dis bgp rout

```
*>      172.16.1.0/24
192.168.35.5
0          400 500i
*>      172.16.2.0/24
192.168.35.5
0          400 500i
*>      172.16.3.0/24
192.168.35.5
0          400 500i
*>      172.16.4.0/24
192.168.35.5
0          400 500i
*>      172.16.5.0/24
```

```

192.168.35.5
0          400 600i
*>        172.16.6.0/24
192.168.35.5
0          400 600i
          *>                                172.16.7.0/24
192.168.35.5
0          400 600i

```

在 R5 上进行路由汇总，并把明细抑制

R5 :

```
bgp 400
```

```
aggregate 172.16.0.0 21 detail-suppressed
```

R3 看到的是汇总路由，只有一个 as 400 的，而原始的 500，600 没有显示出来，存在环路的风险

```

*> 172.16.0.0/21      192.168.35.5      0      400
i

```

如果在汇总的时候加上 as-set,则会继承明细路由的路径属性

R5 :

```
bgp 400
```

```
aggregate 172.16.0.0 21 detail-suppressed as-set
```

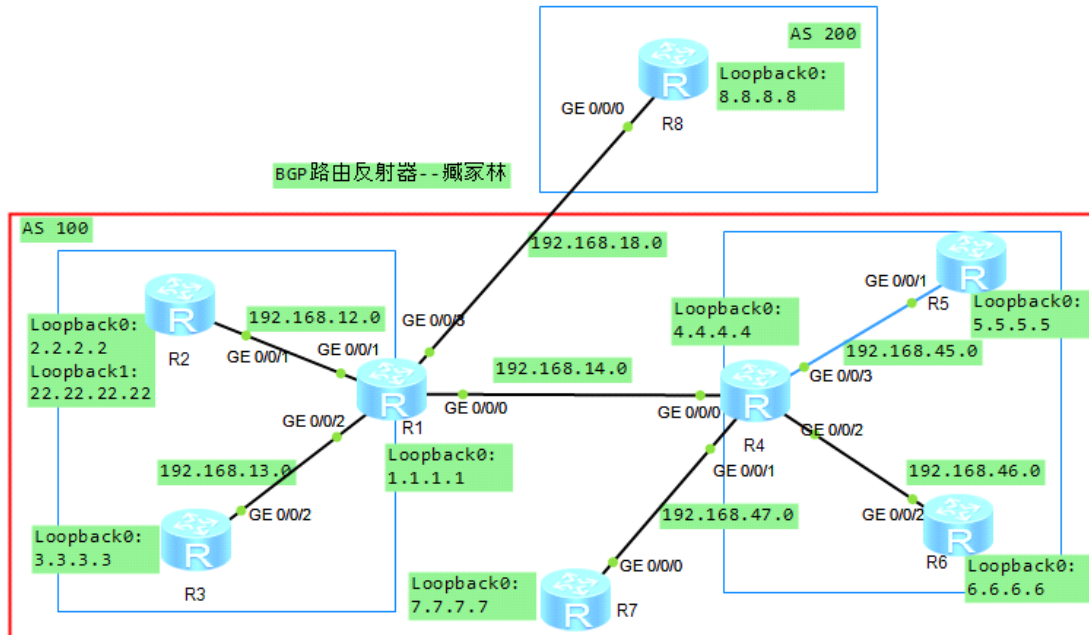
```

*> 172.16.0.0/21      192.168.35.5      0
400 {600 500}i

```

=====

BGP 实验 7 : BGP 路由反射器



IBGP 通过环回口建立，IGP 协议为 OSPF，EBGP 通过物理接口建立

缺省情况下，路由器从它的一个 BGP 对等体那里接收到的路由条目不会被该路由器再传递给其它 IBGP 对等体，这个原则被称为 BGP 水平分割。作用是防止 AS 内部的 BGP 路由环路，因此，在 AS 内部，一般需要每台路由器都运行 BGP 协议并建立全互联的 IBGP 对等体关系。

BGP 路由反射器的使用，可以在很大程度上减少大型 AS 中 IBGP 对等关系的数量并简化相应的配置和维护工作。

分为客户端和非客户端。一个反射器和它所有的客户端一起被称为一个 Cluster，客户端与它的反射器建立 IBGP 对等体关系，客户端之间无需建立 IBGP 对等体关系，非客户端和反射器建立的是 IBGP 对等体关系。

从一个非客户端那里接收到的路由，反射器会将它只传递给所有的客户端，
从一个客户端那里接收到的路由，反射器会将它传递给所有其它的客户端以及非客户端，
从 EBGp 对等体那里接收到的路由，反射器会将它传递给所有客户端和非客户端。

基础配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/1
ip add 192.168.12.1 24
int g0/0/2
ip add 192.168.13.1 24
int g0/0/3
ip add 192.168.18.1 24
int g0/0/0
ip add 192.168.14.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
```

```
ip add 2.2.2.2 24
int loop 1
ip add 22.22.22.22 24
int g0/0/1
ip add 192.168.12.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/2
ip add 192.168.13.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.14.4 24
int g0/0/1
ip add 192.168.47.4 24
int g0/0/2
ip add 192.168.46.4 24
int g0/0/3
```

```
ip add 192.168.45.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/1
ip add 192.168.45.5 24
q
```

```
R6:
undo ter mo
sys
sysname R6
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 6.6.6.6 24
int g0/0/2
ip add 192.168.46.6 24
q
```

```
R7:
undo ter mo
sys
sysname R7
user-interface console 0
idle-timeout 0 0
int loop 0
```

```
ip add 7.7.7.7 24
int g0/0/0
ip add 192.168.47.7 24
q
```

```
R8:
undo ter mo
sys
sysname R8
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 8.8.8.8 24
int g0/0/0
ip add 192.168.18.8 24
q
```

=====

配置 OSPF 协议

```
R1 :
ospf router-id 1.1.1.1
area 0
network 1.1.1.1 0.0.0.0
network 192.168.12.1 0.0.0.0
network 192.168.13.1 0.0.0.0
network 192.168.14.1 0.0.0.0
q
```

```
R2:
ospf router-id 2.2.2.2
area 0
network 2.2.2.2 0.0.0.0
network 192.168.12.2 0.0.0.0
```

q

R3:

ospf router-id 3.3.3.3

area 0

network 3.3.3.3 0.0.0.0

network 192.168.13.3 0.0.0.0

q

R4 :

ospf router-id 4.4.4.4

area 0

network 4.4.4.4 0.0.0.0

network 192.168.14.4 0.0.0.0

network 192.168.45.4 0.0.0.0

network 192.168.46.4 0.0.0.0

network 192.168.47.4 0.0.0.0

q

R5:

ospf router-id 5.5.5.5

area 0

network 5.5.5.5 0.0.0.0

network 192.168.45.5 0.0.0.0

q

R6:

ospf router-id 6.6.6.6

area 0

network 6.6.6.6 0.0.0.0

network 192.168.46.6 0.0.0.0

q

R7:


```
ospf router-id 7.7.7.7
area 0
network 7.7.7.7 0.0.0.0
network 192.168.47.7 0.0.0.0
q
```

=====

配置 BGP 协议

R1 R2 R3 做为一个 Cluster , R4 R5 R6 做为一个 Cluster

peer 1 reflect-client 这条命令的含义是指定 BGP 对等体组 1 中的路由器 R2、R3 为 R1 的客户端，从相反角度来说，也就是 R1 被指定成为 BGP 对等体组 1 中的路由反射器

```
R1:
bgp 100
router-id 1.1.1.1
group 1
peer 2.2.2.2 group 1
peer 3.3.3.3 group 1
peer 1 reflect-client
peer 1 next-hop-local
reflector cluster-id 1
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack 0
peer 4.4.4.4 next-hop-local
peer 192.168.18.8 as-number 200
q
```

```
R2:
bgp 100
router-id 2.2.2.2
peer 1.1.1.1 as-number 100
```

```
peer 1.1.1.1 connect-interface LoopBack 0
network 2.2.2.0 255.255.255.0
network 22.22.22.0 255.255.255.0
q
```

```
R3:
bgp 100
router-id 3.3.3.3
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack 0
network 3.3.3.0 255.255.255.0
q
```

```
R4:
bgp 100
router-id 4.4.4.4
group 2
peer 5.5.5.5 group 2
peer 6.6.6.6 group 2
peer 2 reflect-client
reflector cluster-id 2
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack 0
peer 7.7.7.7 as-number 100
peer 7.7.7.7 connect-interface LoopBack 0
q
```

```
R5:
bgp 100
router-id 5.5.5.5
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack 0
network 5.5.5.0 255.255.255.0
q
```

```
R6:
bgp 100
router-id 6.6.6.6
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack 0
network 6.6.6.0 255.255.255.0
q
```

R7 是非客户端，正常的配置

```
R7:
bgp 100
router-id 7.7.7.7
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack 0
network 7.7.7.0 255.255.255.0
```

配置 R8 与 R1 之间的 EBGP 邻居关系

```
R8:
bgp 200
router-id 8.8.8.8
peer 192.168.18.1 as-number 100
network 8.8.8.0 255.255.255.0
q
```

=====

配置完成后，来查看一下效果

BGP 路由反射器从它的一个客户端收到一个路由之后，会将该路由反射给它的其它客户端、非客户端，以及 EBGP 邻居。

禁止客户机之间的路由反射

undo reflect between-clients

在没有禁止之前，R3 上是有 R2 的路由的

在 R1 上设置

R1:

bgp 100

und reflect between-clients

之后在 R3 上查看，没有 R2 的路由

	Network	NextHop	MED	LocPrf	PrefVa
*>	3.3.3.0/24	0.0.0.0	0		0
*>i	5.5.5.0/24	5.5.5.5	0	100	0
*>i	6.6.6.0/24	6.6.6.6	0	100	0
*>i	8.8.8.0/24	1.1.1.1	0	100	0

[R3]

=====

R1 从 R2 收到 22.22.22.0 ,将它传给了 R3 R4 R8

<R4>display bgp routing-table

*>i	2.2.2.0/24	2.2.2.2	0	100	0	i
*>i	3.3.3.0/24	3.3.3.3	0	100	0	i
*>i	5.5.5.0/24	5.5.5.5	0	100	0	i
*>i	6.6.6.0/24	6.6.6.6	0	100	0	i
*>i	7.7.7.0/24	7.7.7.7	0	100	0	i
*>i	8.8.8.0/24	1.1.1.1	0	100	0	200i
*>i	22.22.22.0/24	2.2.2.2	0	100	0	i

[R4]

R4 将 22.22.22.0 传递给了 R5 R6，但没有传递给 R7，

说明路由反射器会把从非客户端收到的路由传递给客户端，但不会传递给其它非客户端。

<R7>display bgp routing-table

```
*>i 5.5.5.0/24          5.5.5.5          0          100          0          i
*>i 6.6.6.0/24          6.6.6.6          0          100          0          i
*> 7.7.7.0/24          0.0.0.0          0          0          0          i
[R7]
```

路由反射器认为非客户端之间应该是存在 IBGP 对等体关系的，所以路由反射器和非客户端之间依然遵守水平分割原则。

R1 R7 配置为 IBGP 对等体关系。

```
R1:
bgp 100
peer 7.7.7.7 as-number 100
peer 7.7.7.7 connect-interface LoopBack 0
peer 7.7.7.7 next-hop-local
q
```

```
R7:
bgp 100
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack 0
q
```

重新查看 R7 的路由表，就有 22.22.22.0 路由
<R7>display bgp routing-table 22.22.22.0

```
[R7]dis bgp routing-table 22.22.22.0
```

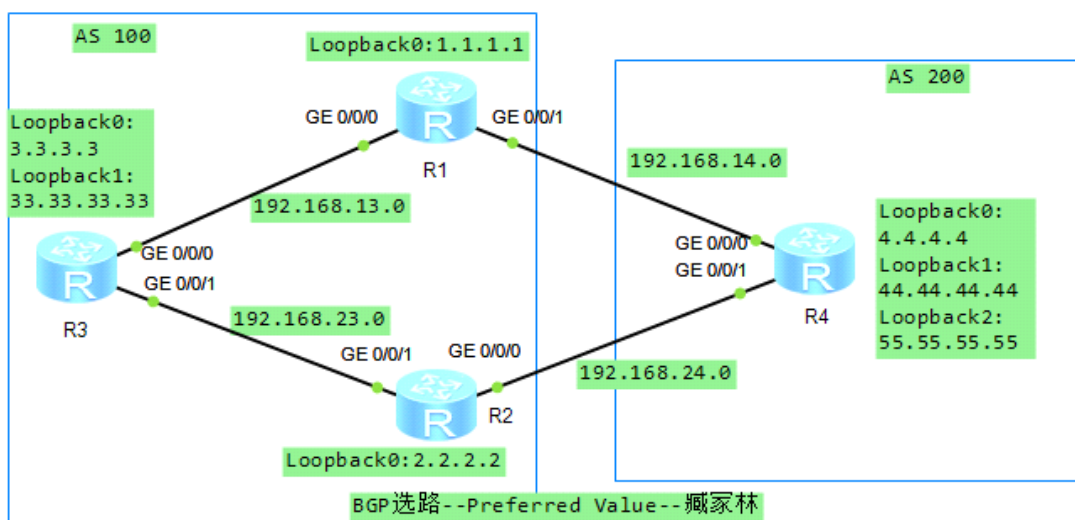
```
BGP local router ID : 7.7.7.7
Local AS number : 100
Paths: 1 available, 1 best, 1 select
BGP routing table entry information of 22.22.22.0/24:
From: 1.1.1.1 (1.1.1.1)
Route Duration: 00h00m01s
```

是 R1 传给 R7 的，不是 R4

R1 从 EBGP 邻居 R8 收到 8.8.8.0 路由之后，将这条路由传递给了 R2 R3 R4，说明路由反射器会把从 EBGP 对等收到的路由传递给它的客户端和非客户端。

=====

BGP 实验 8：BGP 选路—Preferred Value



BG 路由属性的比较过程中，首先要比较的就是路由信息首选值 Preferred Value, 也简称为 PreVal. 值越大，优先级越高，默认为 0，Preferred Value, 不会发送给任何 BGP 邻居，仅作为本地路由器用来选路使用。

IGP 协议为 RIP，全部用环回口建立邻居

基本配置

```
R1:
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.13.1 24
int g0/0/1
ip add 192.168.14.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/1
ip add 192.168.23.2 24
int g0/0/0
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
```

```
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int loop 1
ip add 33.33.33.33 24
int g0/0/0
ip add 192.168.13.3 24
int g0/0/1
ip add 192.168.23.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int loop 1
ip add 44.44.44.44 24
int loop 2
ip add 55.55.55.55 24
int g0/0/0
ip add 192.168.14.4 24
int g0/0/1
ip add 192.168.24.4 24
q
```

在 AS 100 内采用 RIPv2 协议配置 IGP

```
R1 :
rip
version 2
network 192.168.13.0
```



```
network 1.0.0.0
q
```

```
R2 :
rip
version 2
network 192.168.23.0
network 2.0.0.0
q
```

```
R3 :
rip
version 2
network 192.168.13.0
network 192.168.23.0
network 3.0.0.0
q
```

在 R1 R2 R4 上配置静态路由，保证 R1 R2 的 Loopback 0 接口和 R4 的 Loopback 0 能建立 TCP 会话

```
R1 :
ip route-static 4.4.4.0 255.255.255.0 192.168.14.4
```

```
R2 :
ip route-static 4.4.4.0 255.255.255.0 192.168.24.4
```

```
R4 :
ip route-static 1.1.1.0 255.255.255.0 192.168.14.1
ip route-static 2.2.2.0 255.255.255.0 192.168.24.2
```

配置 BGP 协议，用环回口建立邻居

R3 将 33.33.33.0 R4 将 44、55 通告进 BGP 进程中

```
R1 :
bgp 100
router-id 1.1.1.1
```

```
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface loopback 0
peer 2.2.2.2 next-hop-local
peer 3.3.3.3 as-number 100
peer 3.3.3.3 connect-interface loopback 0
peer 3.3.3.3 next-hop-local
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface loopback 0
peer 4.4.4.4 ebgp-max-hop
q
```

```
R2 :
bgp 100
router-id 2.2.2.2
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface loopback 0
peer 1.1.1.1 next-hop-local
peer 3.3.3.3 as-number 100
peer 3.3.3.3 connect-interface loopback 0
peer 3.3.3.3 next-hop-local
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface loopback 0
peer 4.4.4.4 ebgp-max-hop
q
```

```
R3 :
bgp 100
router-id 3.3.3.3
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface loopback 0
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface loopback 0
network 33.33.33.0 24
q
```

```

R4 :
bgp 200
router-id 4.4.4.4
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface loopback 0
peer 1.1.1.1 ebgp-max-hop
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface loopback 0
peer 2.2.2.2 ebgp-max-hop
network 44.44.44.0 24
network 55.55.55.0 24
q

```

配置完成后，查看邻居的建立

<R1>display bgp peer

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
2.2.2.2	4	100	3	6	0	00:01:46	Established
3.3.3.3	4	100	3	5	0	00:00:51	Established
4.4.4.4	4	200	4	4	0	00:00:11	Established

[R1]_

<R2>display bgp peer

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
1.1.1.1	4	100	6	6	0	00:02:21	Established
3.3.3.3	4	100	4	6	0	00:01:19	Established
4.4.4.4	4	200	5	7	0	00:00:32	Established

[R2]

查看 R3 的路由表，去往 44 55 的路由有两条，但 R3 都优选下一跳为 1.1.1.1 的路由，属性相同，BGP 最终选择 router-id 较小的路由器 R1 做为最佳路由

<R3>display bgp routing-table

```

Total Number of Routes: 5
      Network                NextHop          MED           LocPrf        PrefVal
*> 33.33.33.0/24             0.0.0.0              0              0              0
*>i 44.44.44.0/24             1.1.1.1              0             100             0
* i 2.2.2.2                  2.2.2.2              0             100             0
*>i 55.55.55.0/24             1.1.1.1              0             100             0
* i 2.2.2.2                  2.2.2.2              0             100             0
[R3]

```

无论是通过 EBGP 邻居还是 IBGP 邻居学习的到 BGP 路由条目，在 BGP 路由表中的 PrefVal 都显示为缺省值 0

=====

修改 PrefVal 值

修改 PrefVal 值，使 R3 去 55.55.55.0 从 R2 走

R3:

```

ip ip-prefix 1 permit 55.55.55.0 24
route-policy 10 permit node 1
if-match ip-prefix 1
apply preferred-value 99
route-policy 10 permit node 2

```

bgp 100

peer 2.2.2.2 route-policy 10 import

配置完成后，查看 R3 的 BGP 路由表

<R3>display bgp routing-table

```

*> 33.33.33.0/24             0.0.0.0              0              0              0
*>i 44.44.44.0/24             1.1.1.1              0             100             0
* i 2.2.2.2                  2.2.2.2              0             100             0
*>i 55.55.55.0/24             2.2.2.2              0             100             99
* i 1.1.1.1                  1.1.1.1              0             100             0
[R3]

```

在 R3 上修改之后，R1 R2 上的 PrefVal 值不变，说明了 Pref

Val 值只作为本地路由器用来选路使用，并不传递给任何 BGP 邻居。

<R1>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>i	33.33.33.0/24	3.3.3.3	0	100	0
*>	44.44.44.0/24	4.4.4.4	0		0
* i		2.2.2.2	0	100	0
*>	55.55.55.0/24	4.4.4.4	0		0
* i		2.2.2.2	0	100	0

[R1]

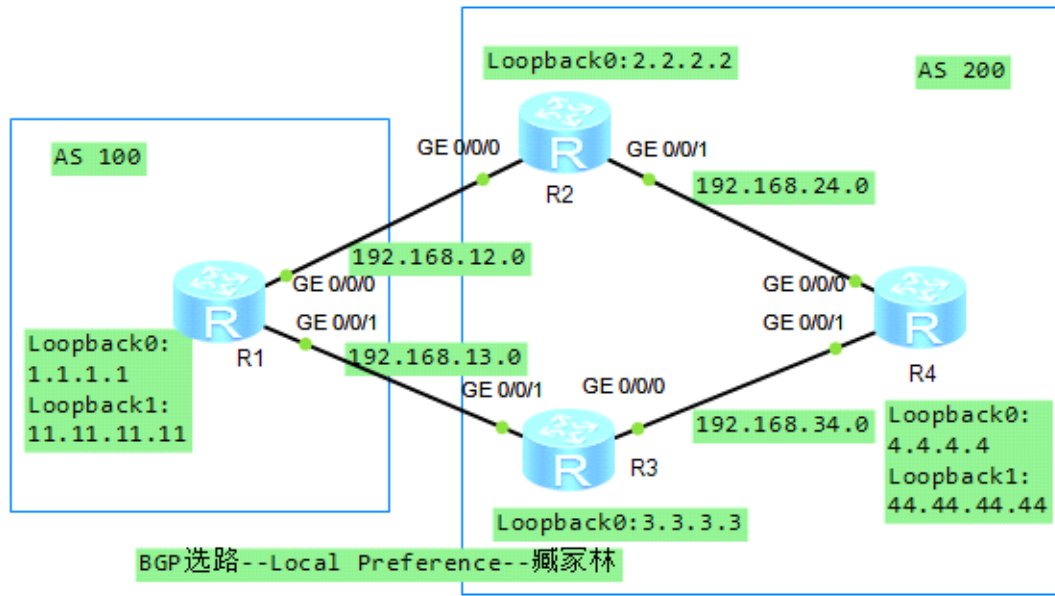
在 R2 上查看 <R2>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>i	33.33.33.0/24	3.3.3.3	0	100	0
*>	44.44.44.0/24	4.4.4.4	0		0
* i		1.1.1.1	0	100	0
*>	55.55.55.0/24	4.4.4.4	0		0
* i		1.1.1.1	0	100	0

[R2]

= = = = = = = =

BGP 实验 9 : BGP 选路—Local Preference



IBGP 通过环回口建立，IGP 协议为 OSPF，EBGP 通过物理接口建立

BGP 首先比较路由信息的首选值，如何 PrefVal 相同，就会比较本地优先级（Local Preference）牌佛润斯 属性。越大，越优先，默认为 100

Local Preference 可以用于选择流量离开 AS 时的最佳路由，也就是控制流量从哪个出口离开 AS，选择值高的路由。Local Preference 只在 IBGP 对等体之间进行通告，EBGP 对等体之间传递 BGP 路由时不携带 Local Preference 属性。当从 EBG P 对等体收到的路由 Local Preference 值为空时，接收路由器会使用 100 作为默认值。

Local-Pref 只在 IBGP 中传递，用来控制出去流量的走向
MED 能通过 EBGP 传一个 AS，用来控制对端 AS 入流量的走向

举个例子，A 公司老板让 A 公司员工做某件事，就相当于 Local-Pref。B 公司老板让 A 公司员工做某件事就相当于 MED。那么如果 A 公司老板让 A 公司员工做一件事，同时 B 公司老板又让 A 公司员工做相反的事会怎么样呢？
答案很明显，Local-Pref 优先

基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int loop 1
ip add 11.11.11.11 24
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.13.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
```

```
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.13.3 24
int g0/0/0
ip add 192.168.34.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int loop 1
ip add 44.44.44.44 24
int g0/0/1
ip add 192.168.24.4 24
int g0/0/0
ip add 192.168.34.4 24
```


q

完成 OSPF 和 BGP 协议的基本配置 , AS 200 中的路由器运行 OSPF

R2:

```
ospf router-id 2.2.2.2
area 0
network 192.168.24.2 0.0.0.0
network 2.2.2.2 0.0.0.0
```

q

R3:

```
ospf router-id 3.3.3.3
area 0
network 192.168.34.3 0.0.0.0
network 3.3.3.3 0.0.0.0
```

q

R4:

```
ospf router-id 4.4.4.4
area 0
network 192.168.24.4 0.0.0.0
network 192.168.34.4 0.0.0.0
network 4.4.4.4 0.0.0.0
```

q

配置完成后在 R4 上查看 OSPF 邻居关系

<R4>display ospf peer brief

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	2.2.2.2	Full
0.0.0.0	GigabitEthernet0/0/0	3.3.3.3	Full

[R4]

=====

配置 BGP 协议

R1:

```
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
peer 192.168.13.3 as-number 200
network 1.1.1.0 24
network 11.11.11.0 24
q
```

R2:

```
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 3.3.3.3 as-number 200
peer 3.3.3.3 connect-interface loop 0
peer 3.3.3.3 next-hop-local
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface loop 0
peer 4.4.4.4 next-hop-local
q
```

R3:

```
bgp 200
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
peer 2.2.2.2 as-number 200
peer 2.2.2.2 connect-interface loop 0
peer 2.2.2.2 next-hop-local
```

```
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface loop 0
peer 4.4.4.4 next-hop-local
q
```

```
R4:
bgp 200
router-id 4.4.4.4
peer 2.2.2.2 as-number 200
peer 2.2.2.2 connect-interface loop 0
peer 3.3.3.3 as-number 200
peer 3.3.3.3 connect-interface loop 0
network 44.44.44.0 24
q
```

查看 BGP 邻居关系是否建立成功

<R1>display bgp peer <R4>display bgp peer

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.2	4	200	4	7	0	00:01:31	Established
192.168.13.3	4	200	3	7	0	00:00:25	Established

[R1]

在 R4 上查看 BGP 路由表

<R4>display bgp routing-table

Total Number of Routes: 5

Network	NextHop	MED	LocPrf	PrefVal
*>i 1.1.1.0/24	2.2.2.2	0	100	0
* i	3.3.3.3	0	100	0
*>i 11.11.11.0/24	2.2.2.2	0	100	0
* i	3.3.3.3	0	100	0
*> 44.44.44.0/24	0.0.0.0	0		0

[R4]

可以看到 1.1.1.0 11.11.11.0 两条路由的 Local

Preference 值默认为 100，本地产生的 44.44.44.0 值为空

<R2>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>	1.1.1.0/24	192.168.12.1	0		0
* i		3.3.3.3	0	100	0
*>	11.11.11.0/24	192.168.12.1	0		0
* i		3.3.3.3	0	100	0
*>i	44.44.44.0/24	4.4.4.4	0	100	0

[R2]

R2 收到的 1.1.1.0 路由，从 EBGp 邻居 R1 收到的 Local Preference 值为空，从 IBGP 邻居 R3 收到的值为 100.由此可见 Local Preference 值不会通告给 EBGp 对等体

=====

修改 Local Preference 值

<R3>display default-parameter bgp 查看 BGP 协议的默认参数

```
Local preference           : 255
BGP connect-retry          : 32s
BGP holdtime               : 180s
BGP keepAlive              : 60s
EBGP route-update-interval: 30s
IBGP route-update-interval: 15s
Default local-preference   : 100
Default MED                 : 0
```

<R4>display bgp routing-table

```

Total Number of Routes: 5
      Network          NextHop      MED      LocPrf    PrefVal
* > i  1.1.1.0/24      2.2.2.2        0        100       0
* i    3.3.3.3        0        100       0
* > i  11.11.11.0/24   2.2.2.2        0        100       0
* i    3.3.3.3        0        100       0
* >    44.44.44.0/24   0.0.0.0        0              0
[R4]

```

R4 选择 router-id 较小的 R2 到达 11.11.11.0
 现在修改一下，让他从 R3 走

R3:
 bgp 200
 default local-preference 200

<R4>display bgp routing-table 在 R4 上查看效果

```

      Network          NextHop      MED      LocPrf    PrefVal
* > i  1.1.1.0/24      3.3.3.3        0        200       0
* > i  11.11.11.0/24   3.3.3.3        0        200       0
* >    44.44.44.0/24   0.0.0.0        0              0
[R4]

```

<R4>tracert -a 44.44.44.44 11.11.11.11 是从 R3 走的路由

```
[R4]tracert -a 44.44.44.44 11.11.11.11
```

```

  traceroute to 11.11.11.11(11.11.11.11)
, max hops: 30 ,packet length: 40,press

```

```
1 192.168.34.3 70 ms 30 ms 40 ms
```

```
2 192.168.13.1 80 ms 80 ms 80 ms
```

=====

使用 route-policy 修改 Local Preference 值

<R4>tracert -a 44.44.44.44 1.1.1.1 现在 R4 到 1.1.1.1 也是从 R3 走

修改 R4 去往 1.1.1.1 从 R2 走，去 11.11.11.0 从 R3 走

R2:

```
ip ip-prefix 1 permit 1.1.1.0 24
```

```
route-policy 10 permit node 1
```

```
if-match ip-prefix 1
```

```
apply local-preference 299
```

```
route-policy 10 permit node 2
```

```
bgp 200
```

```
peer 192.168.12.1 route-policy 10 import
```

在 R2 接收 R1 所传递的路由信息的 import 方向上调用路由策略。

<R2>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>	1.1.1.0/24	192.168.12.1	0	299	0
*>i	11.11.11.0/24	3.3.3.3	0	200	0
*		192.168.12.1	0		0
*>i	44.44.44.0/24	4.4.4.4	0	100	0

[R2]

<R4>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>i	1.1.1.0/24	2.2.2.2	0	299	0
*>i	11.11.11.0/24	3.3.3.3	0	200	0
*>	44.44.44.0/24	0.0.0.0	0		0

[R4]

<R4>tracert -a 44.44.44.44 1.1.1.1

R4 去往 1.1.1.1 从 R2 走，去往 11.11.11.11 从 R3 走

[R4]tracert -a 44.44.44.44 1.1.1.1

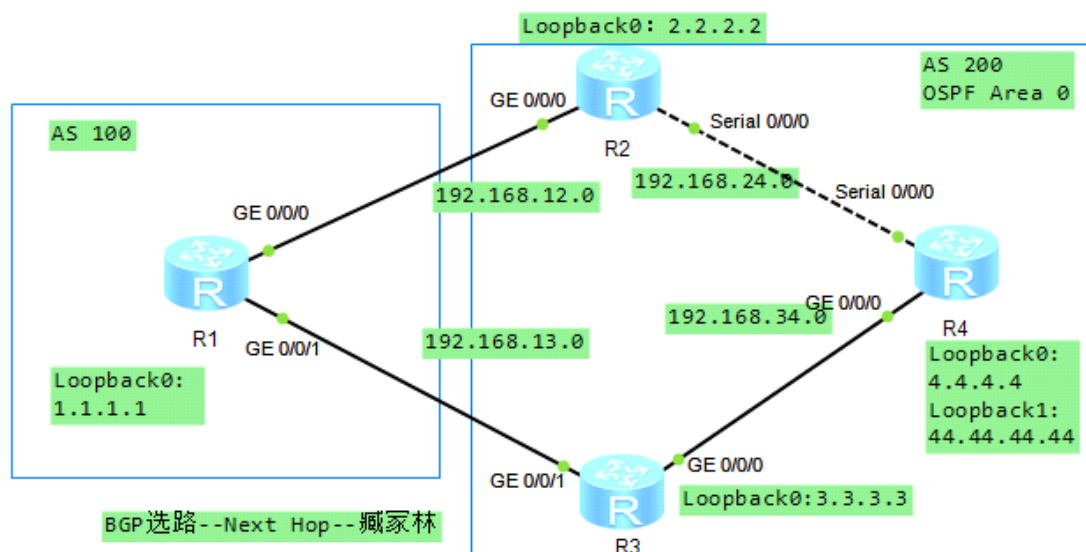
traceroute to 1.1.1.1(1.1.1.1), max
o break

1 192.168.24.2 40 ms 50 ms 50 ms

2 192.168.12.1 40 ms 80 ms 60 ms

= = = = =

BGP 实验 10 : BGP 选路—Next Hop



Next Hop 属性记录了去往目标网络所对应的下一跳 IP 地址。BGP 在比较 Next Hop 属性时，会优选去往 Next Hop 属性中 IP 地址的 IGP 路由协议路径开销最小的路由。

IBGP 通过环回口建立，IGP 协议为 OSPF，EBGP 通过物理接口建立

基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.13.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int s0/0/0
```



```
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.13.3 24
int g0/0/0
ip add 192.168.34.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int loop 1
ip add 44.44.44.44 24
int s0/0/0
ip add 192.168.24.4 24
int g0/0/0
ip add 192.168.34.4 24
q
```

配置 IGP 协议 OSPF

R2:
ospf router-id 2.2.2.2
area 0
network 192.168.24.2 0.0.0.0
network 2.2.2.2 0.0.0.0
q

R3:
ospf router-id 3.3.3.3
area 0
network 192.168.34.3 0.0.0.0
network 3.3.3.3 0.0.0.0
q

R4:
ospf router-id 4.4.4.4
area 0
network 192.168.24.4 0.0.0.0
network 192.168.34.4 0.0.0.0
network 4.4.4.4 0.0.0.0
q

完成后在 R4 上查看 OSPF 邻居关系

<R4>display ospf peer brief

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial0/0/0	2.2.2.2	Full
0.0.0.0	GigabitEthernet0/0/0	3.3.3.3	Full

[R4]

=====

配置 BGP 路由协议

R1:

bgp 100

router-id 1.1.1.1

peer 192.168.12.2 as-number 200

peer 192.168.13.3 as-number 200

network 1.1.1.0 24

q

R2:

bgp 200

router-id 2.2.2.2

peer 192.168.12.1 as-number 100

peer 3.3.3.3 as-number 200

peer 3.3.3.3 connect-interface loop 0

peer 3.3.3.3 next-hop-local

peer 4.4.4.4 as-number 200

peer 4.4.4.4 connect-interface loop 0

peer 4.4.4.4 next-hop-local

q

R3:

bgp 200

router-id 3.3.3.3

peer 192.168.13.1 as-number 100

peer 2.2.2.2 as-number 200

peer 2.2.2.2 connect-interface loop 0

peer 2.2.2.2 next-hop-local

peer 4.4.4.4 as-number 200

peer 4.4.4.4 connect-interface loop 0

peer 4.4.4.4 next-hop-local

q

R4:

bgp 200

```

router-id 4.4.4.4
peer 2.2.2.2 as-number 200
peer 2.2.2.2 connect-interface loop 0
peer 3.3.3.3 as-number 200
peer 3.3.3.3 connect-interface loop 0
network 44.44.44.0 24
q

```

配置完成后，在 R2 上查看 BGP 邻居关系

```
<R2>display bgp peer
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
3.3.3.3	4	200	4	5	0	00:01:24	Established
4.4.4.4	4	200	4	5	0	00:01:10	Established
192.168.12.1	4	100	5	4	0	00:01:32	Established

[R2]

在 R1 上验证从 1.1.1.1 去往 44.44.44.44 报文所经过的路径

```
<R1>tracert -a 1.1.1.1 44.44.44.44，是通过 R2 到达的
```

```
[R1]tracert -a 1.1.1.1 44.44.44.44
```

```

  traceroute to 44.44.44.44(44.44.44.44)
  CTRL_C to break

```

```
1 192.168.12.2 50 ms 60 ms 50 ms
```

```
2 192.168.24.4 90 ms 90 ms 100 ms
```

<R4>tracert -a 44.44.44.44 1.1.1.1，R4 是通过 R3 到达 R1

```
[R4]tracert -a 44.44.44.44 1.1.1.1
```

```
tracert to 1.1.1.1(1.1.1.1), max
o break
```

```
1 192.168.34.3 30 ms 60 ms 70 ms
```

```
2 192.168.13.1 90 ms 110 ms 130 ms
```

通信双方的往返报文选用不同路径的现象称为不对称路由。对于某些特定的应用，以及部署了某些特别的安全设备和安全策略的情况下，不对称路由的存在可能会导致通过中断的现象。

```
<R4>display bgp routing-table
```

	Network	NextHop	MED	LocPrf
*>i	1.1.1.0/24	3.3.3.3	0	100
* i		2.2.2.2	0	100
*>	44.44.44.0/24	0.0.0.0	0	

```
[R4]
```

两条路由都来自 IBGP 对等体，所以需要比较两条路由信息中去往 Next Hop 地址的 IGP 开销，并选择开销更小的路由。

```
<R4>display ip routing-table
```

1.1.1.0/24	IBGP	255	0
2.2.2.2/32	OSPF	10	1562
3.3.3.3/32	OSPF	10	1
4.4.4.0/24	Direct	0	0
4.4.4.4/32	Direct	0	0
44.44.44.0/24	Direct	0	0
44.44.44.44/32	Direct	0	0

去往 3.3.3.3 开销为 1，去往 2.2.2.2 开销为 1562，所以 BGP 选择 R3

=====

修改接口开销

为了避免不对称路由，修改 R4 g0/0/0 开销，使它比 s0/0/0 还要大

R4:

```
int g0/0/0
```

```
ospf cost 2000
```

nation/Mask	Proto	Pre	Cost
1.1.1.0/24	IBGP	255	0
2.2.2.2/32	OSPF	10	1562
3.3.3.3/32	OSPF	10	2000

配置完成后在 R4 上看一下，<R4>display bgp routing-table

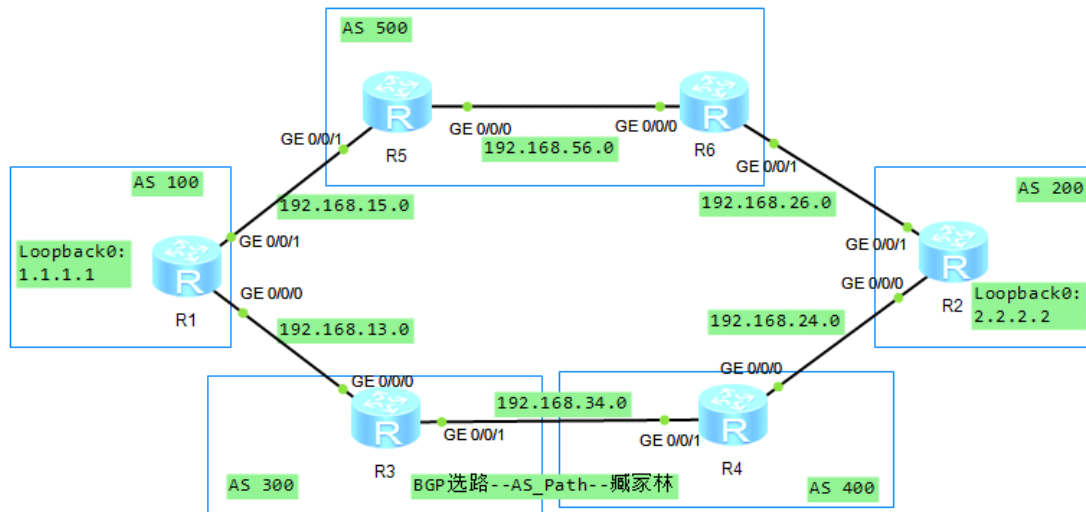
	Network	NextHop	MED	LocPrf	PrefVal
*>i	1.1.1.0/24	2.2.2.2	0	100	0
* i		3.3.3.3	0	100	0
*>	44.44.44.0/24	0.0.0.0	0		0

[R4]

现在 R1 去 R4，R4 去 R1 都从 R2 经过。

=====

BGP 实验 11 : BGP 选路—AS-Path



全部使用物理接口建立邻居关系

AS_Path 属性顺序记录了某条 BGP 路由所经过的 AS 信息，BGP 在比较 AS_Path 属性后，会优选 AS_Path 长度较短的那条路由。另外，AS_Path 还可以用来防止路由之间的环路。当路由器从 EBGP 邻居收到 BGP 路由时，如果该路由的 AS_Path 中包含了自己的 AS 编号，则该路由将会直接丢弃。

基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/1
ip add 192.168.15.1 24
```

```
int g0/0/0
ip add 192.168.13.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/1
ip add 192.168.26.2 24
int g0/0/0
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 192.168.13.3 24
int g0/0/1
ip add 192.168.34.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
```



```
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 192.168.24.4 24
int g0/0/1
ip add 192.168.34.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int g0/0/1
ip add 192.168.15.5 24
int g0/0/0
ip add 192.168.56.5 24
q
```

```
R6:
undo ter mo
sys
sysname R6
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 192.168.56.6 24
int g0/0/1
ip add 192.168.26.6 24
q
```

配置 BGP 路由协议

```
R1:
```

```
bgp 100
router-id 1.1.1.1
peer 192.168.15.5 as-number 500
peer 192.168.13.3 as-number 300
network 1.1.1.0 255.255.255.0
q
```

```
R2:
bgp 200
router-id 2.2.2.2
peer 192.168.26.6 as-number 500
peer 192.168.24.4 as-number 400
network 2.2.2.0 255.255.255.0
q
```

```
R3:
bgp 300
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
peer 192.168.34.4 as-number 400
q
```

```
R4:
bgp 400
router-id 4.4.4.4
peer 192.168.34.3 as-number 300
peer 192.168.24.2 as-number 200
q
```

```
R5:
bgp 500
router-id 5.5.5.5
peer 192.168.15.1 as-number 100
peer 192.168.56.6 as-number 500
```

```
peer 192.168.56.6 next-hop-local
q
```

```
R6:
bgp 500
router-id 6.6.6.6
peer 192.168.26.2 as-number 200
peer 192.168.56.5 as-number 500
peer 192.168.56.5 next-hop-local
q
```

配置完成后，查看 BGP 邻居关系的建立

```
<R1>display bgp peer
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.13.3	4	300	4	5	0	00:00:17	Established
192.168.15.5	4	500	2	6	0	00:00:02	Established

[R1]

```
<R1>display bgp routing-table
```

Total Number of Routes: 3

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 1.1.1.0/24	0.0.0.0	0		0	i
*> 2.2.2.0/24	192.168.15.5			0	500 200i
* 2.2.2.0/24	192.168.13.3			0	300 400 200i

[R1]

可以看到 R1 去往 2.2.2.0 存在两条路由，但优选 R5
因为 R3 的 AS-Path 为 300 400 200，长度为 3，R5 为 500 200，长度为 2
于是 R1 选择了 AS-Path 长度小的 R5

```
<R1>tracert -a 1.1.1.1 2.2.2.2
```

```
[R1]tracert -a 1.1.1.1 2.2.2.2
```

```
  traceroute to 2.2.2.2(2.2.2.2), max hops:
o break
```

```
 1 192.168.15.5 60 ms  40 ms  60 ms
```

```
 2 192.168.56.6 90 ms  80 ms  90 ms
```

```
 3 192.168.26.2 160 ms 110 ms 160 ms
```

```
[R1]
```

R2 去往 1.1.1.0 也是同样的原因，选择了从 R6 经过

=====

修改 AS-Path 属性控制 BGP 选路，有两种修改方式：additive 和 overwrite

使用 route-policy 对 R1 接收到的 AS 500 关于 2.2.2.0 的路由 AS-Path 修改

R1:

```
ip ip-prefix asp permit 2.2.2.0 24
```

```
route-policy 10 permit node 1
```

```
if-match ip-prefix asp
```

```
apply as-path 500 500 additive
```

```
route-policy 10 permit node 2
```

```
bgp 100
```

```
peer 192.168.15.5 route-policy 10 import
```

```
q
```

<R1>display bgp routing-table 在 R1 上来查看修改之后的效果

```
Total Number of Routes: 3
Network           NextHop           MED           LocPrf        PrefVal Path/Ogn
*> 1.1.1.0/24      0.0.0.0           0
*> 2.2.2.0/24      192.168.13.3      0             0             300 400 200i
*                  192.168.15.5      0             0             500 500 500 200i
[R1]
```

现在 R1 到达 2.2.2.0 优先从 R3 走

=====

修改 R2 ，让 R2 到达 R1 优先从 R4 R3 走

R2:

ip ip-prefix asp permit 1.1.1.0 24

route-policy 10 permit node 1

if-match ip-prefix asp

apply as-path 300 500 500 100 overwrite

Warning: The AS-Path lists of routes to which this route-policy is applied will be overwritten. Continue?

[Y/N] y

使用 overwritten 意味着将覆盖原有的 AS-Path

route-policy 10 permit node 2

bgp 200

peer 192.168.26.6 route-policy 10 import

q

<R2>display bgp routing-table 查看效果

```
Total Number of Routes: 3
Network           NextHop           MED           LocPrf        PrefVal Path/Ogn
*> 1.1.1.0/24      192.168.24.4      0             0             400 300 100i
*                  192.168.26.6      0             0             300 500 500 100i
*> 2.2.2.0/24      0.0.0.0           0             0             1
[R2]
```

<R2>tracert -a 2.2.2.2 1.1.1.1

```
[R2]tracert -a 2.2.2.2 1.1.1.1
```

```
tracert to 1.1.1.1(1.1.1.1), max hops:  
o break
```

```
1 192.168.24.4 30 ms 70 ms 60 ms
```

```
2 192.168.34.3 120 ms 80 ms 130 ms
```

```
3 192.168.13.1 120 ms 110 ms 130 ms
```

```
<R1>tracert -a 1.1.1.1 2.2.2.2
```

```
[R1]tracert -a 1.1.1.1 2.2.2.2
```

```
tracert to 2.2.2.2(2.2.2.2), max hops  
o break
```

```
1 192.168.13.3 60 ms 40 ms 30 ms
```

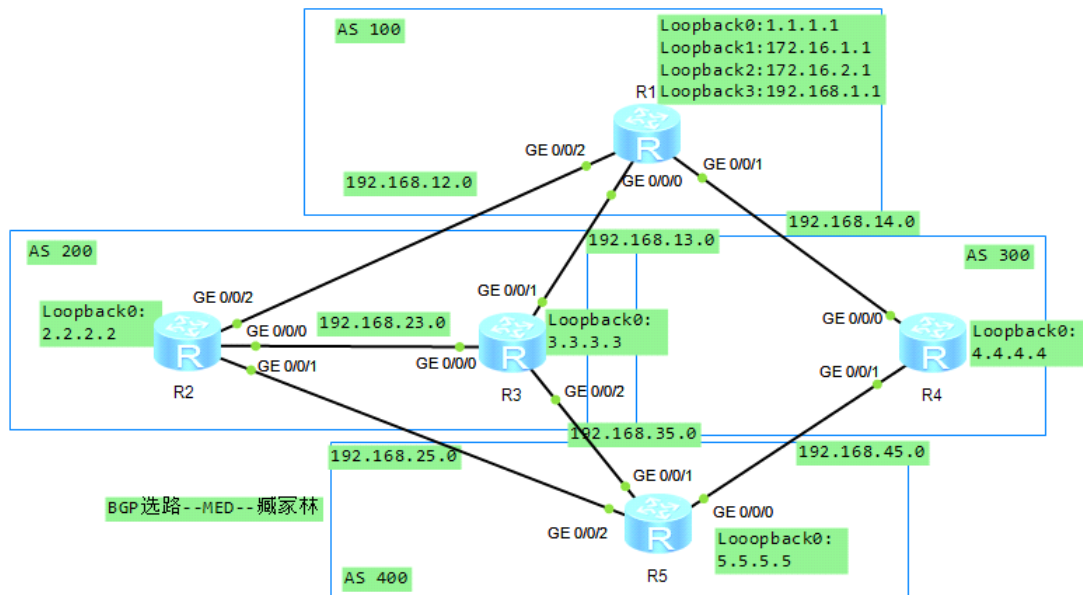
```
2 192.168.34.4 80 ms 120 ms 90 ms
```

```
3 192.168.24.2 140 ms 160 ms 90 ms
```

都是从 R3 R4 经过，实现了所需的路径控制

=====

BGP 实验 12 : BGP 选路—MED



全网用物理接口建立 BGP 邻居

MED(Multi-exit-disc)也称为多出口鉴别器，它是一个 4 字节的整数，默认 MED 值为 0，MED 值越小，表明相应的路由优先级越高。

MED 属性主要作用是用来控制来自邻居 AS 的流量从哪个入口进入到本 AS 中。

默认，只有去往同一个目标网络的多条路由均来自同一个邻居 AS 时，BGP 才比较 MED 值，配置命令 `compare-different-as-med` 后，则会比较来自不同邻居 AS 的目标网络相同的 BGP 路由的 MED 值。

MED 属性只会影响相邻的两个 AS，收到 MED 属性的 AS 不会把此属性再传给别的 AS。

=====

deterministic-med (BGP) 理论

deterministic-med 命令后，路由器基于邻居 AS 和 MED 分类路径，以保证任何时候都用相同的方法分类路径。这样可以产生确定性的最好路径。

未使能 deterministic-med 时，会按照路由接收的顺序依次进行比较，最终选路的结果和路由的接收顺序是相关的。

使能 deterministic-med 功能，在对从多个不同 AS 收到的相同前缀的路由进行选路时，首先会按路由的 AS_Path 最左边的 AS 号进行分组。在组内进行比较后，再用组中的优选路由和其他组中的优选路由进行比较，消除了选路的结果和路由接收顺序的相关性。

例如：假设在某台路由器上存在如下三条 BGP 路由：

- Route A1: AS(PATH) 12, med 100, igp metric 13, internal, rid 4.4.4.4
- Route A2: AS(PATH) 12, med 150, igp metric 11, internal, rid 5.5.5.5
- Route B: AS(PATH) 3, med 0, igp metric 12, internal, rid 6.6.6.6

当路由收来的顺序为：Route A1、Route A2、Route B 时，先比较 A1、A2。因为 Route A1、Route A2 的最左 AS 相同所以优选 MED 较小的路由 Route A1。再比较 Route A1 和 Route B，因为 Route A1、Route B 的最左 AS 不相同，在未配置 compare-different-as-med (BGP) 命令情况下，不能比较 MED。优选 IGP Metric 较小的路由 Route B。

当路由收来的顺序为：Route A2、Route B、Route A1 时，先比较 Route A2、Route B，因为 Route B 和 Route A2 的最左 AS 不相同，在未配置 compare-different-as-med (BGP) 命

令情况下，优选 IGP Metric 较小的路由 Route A2。再比较 Route A2、Route A1，因为 Route A2 和 Route A1 的最左 AS 相同，所以优选 MED 较小的路由 Route A1。

从上面的分析可以看到，未使能 deterministic-med 时，最终优选的路由和路由接收的顺序相关。而使能了 deterministic-med 后，就会消除选路的结果和路由接收顺序的相关性。

=====

基础配置

R1:

```
undo terminal
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int loop 1
ip add 172.16.1.1 24
int loop 2
ip add 172.16.2.1 24
int loop 3
ip add 192.168.1.1 24
int g0/0/2
ip add 192.168.12.1 24
int g0/0/0
ip add 192.168.13.1 24
int g0/0/1
ip add 192.168.14.1 24
```

q

R2:

undo ter mo

sys

sysname R2

user-interface console 0

idle-timeout 0 0

int loop 0

ip add 2.2.2.2 24

int g0/0/2

ip add 192.168.12.2 24

int g0/0/0

ip add 192.168.23.2 24

int g0/0/1

ip add 192.168.25.2 24

q

R3:

undo ter mo

sys

sysname R3

user-interface console 0

idle-timeout 0 0

int loop 0

ip add 3.3.3.3 24

int g0/0/1

ip add 192.168.13.3 24

int g0/0/0

ip add 192.168.23.3 24

int g0/0/2

ip add 192.168.35.3 24

q

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.14.4 24
int g0/0/1
ip add 192.168.45.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/2
ip add 192.168.25.5 24
int g0/0/1
ip add 192.168.35.5 24
int g0/0/0
ip add 192.168.45.5 24
q
```

配置 BGP 路由协议，使用直连物理接口建立 BGP 邻居

```
R1 :
bgp 100
router-id 1.1.1.1
```

```
peer 192.168.12.2 as-number 200
peer 192.168.13.3 as-number 200
peer 192.168.14.4 as-number 300
network 1.1.1.0 255.255.255.0
network 172.16.1.0 255.255.255.0
network 172.16.2.0 255.255.255.0
network 192.168.1.0 255.255.255.0
q
```

```
R2 :
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 192.168.23.3 as-number 200
peer 192.168.23.3 next-hop-local
peer 192.168.25.5 as-number 400
network 2.2.2.0 255.255.255.0
q
```

```
R3 :
bgp 200
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
peer 192.168.23.2 as-number 200
peer 192.168.23.2 next-hop-local
peer 192.168.35.5 as-number 400
network 3.3.3.0 255.255.255.0
q
```

```
R4 :
bgp 300
router-id 4.4.4.4
peer 192.168.14.1 as-number 100
peer 192.168.45.5 as-number 400
```

```
network 4.4.4.0 255.255.255.0
```

```
q
```

R5 :

```
bgp 400
```

```
router-id 5.5.5.5
```

```
peer 192.168.25.2 as-number 200
```

```
peer 192.168.35.3 as-number 200
```

```
peer 192.168.45.4 as-number 300
```

```
network 5.5.5.0 255.255.255.0
```

```
q
```

配置完成后，查看邻居的建立情况

```
<R1>display bgp peer
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.2	4	200	4	11	0	00:00:56	Established
192.168.13.3	4	200	4	11	0	00:00:48	Established
192.168.14.4	4	300	3	10	0	00:00:00	Established

```
[R1]
```

```
<R2>display bgp routing-table
```

Total Number of Routes: 15		NextHop	MED	LocPrf	PrefVal
	Network				
*>	1.1.1.0/24	192.168.12.1	0		0
* i		192.168.23.3	0	100	0
*>	2.2.2.0/24	0.0.0.0	0		0
*> i	3.3.3.0/24	192.168.23.3	0	100	0
*>	4.4.4.0/24	192.168.12.1			0
*		192.168.25.5			0
* i		192.168.23.3		100	0
*>	5.5.5.0/24	192.168.25.5	0		0
* i		192.168.23.3	0	100	0
*>	172.16.1.0/24	192.168.12.1	0		0
* i		192.168.23.3	0	100	0
*>	172.16.2.0/24	192.168.12.1	0		0
* i		192.168.23.3	0	100	0
*>	192.168.1.0	192.168.12.1	0		0
* i		192.168.23.3	0	100	0

```
[R2]
```

路由的 MED 值为 0，如果接收到的路由条目经过一个 AS 进

行中转，那么 MED 值将会丢失，设置为空。MED 值为空实际上等同于值为 0

=====

控制来自同一 AS 的路由选择最佳路径

根据 BGP 选路机制，EBGP 路由优于 IBGP 路由的原则，R2 去往 R1 的环回口选择 R1 的 g0/0/2，R3 去 R1 的环回口选择 R1 的 g0/0/0。

现在要求 R2 R3 访问 172.16.1.0 从 R1 的 g0/0/2 接口进入，也就是经过 R2

R2 R3 访问 172.16.2.0 从 R1 的 g0/0/0 接口进入，也就是经过 R3

前缀列表匹配要修改的 MED 路由

R1:

```
ip ip-prefix 1 permit 172.16.1.0 24
```

```
ip ip-prefix 2 permit 172.16.2.0 24
```

```
route-policy R2 permit node 1
```

```
if-match ip-prefix 1
```

```
apply cost 100
```

```
route-policy R2 permit node 2
```

```
if-match ip-prefix 2
```

```
apply cost 200
```

```
route-policy R2 permit node 3
```

R1 与 R2 之间调用 route-policy R2

```
bgp 100
```

```
peer 192.168.12.2 route-policy R2 export
```

再创建 route-policy R3 , 与上面的 cost 相反

```
route-policy R3 permit node 1
if-match ip-prefix 1
apply cost 200
route-policy R3 permit node 2
if-match ip-prefix 2
apply cost 100
route-policy R3 permit node 3
```

R1 与 R3 之间调用 route-policy R3

```
bgp 100
peer 192.168.13.3 route-policy R3 export
```

配置完成后 , 在 R2 R3 上查看 BGP 路由表

<R2>display bgp routing-table

```
*> 172.16.1.0/24 192.168.12.1 100 0 100i
*>i 172.16.2.0/24 192.168.23.3 100 100 0 100i
* 192.168.12.1 200 0 100i
*> 192.168.1.0 192.168.12.1 0 0 100i
* i 192.168.23.3 0 100 0 100i
[R2]
```

```
*>i 172.16.1.0/24 192.168.23.2 100 100 0
* 192.168.13.1 200 0
*> 172.16.2.0/24 192.168.13.1 100 0
*> 192.168.1.0 192.168.13.1 0 0
* i 192.168.23.2 0 100 0
[R3]
```

可以路由跟踪，查看是否符合要求：

R2 R3 访问 172.16.1.0 从 R1 的 g0/0/2 接口进入，也就是经过 R2

R2 R3 访问 172.16.2.0 从 R1 的 g0/0/0 接口进入，也就是经过 R3

```
<R2>tracert -a 2.2.2.2 172.16.1.1
```

```
<R2>tracert -a 2.2.2.2 172.16.2.1
```

=====

控制来自不同 AS 且去往同一目标网络的数据流量选路

```
<R5>display bgp routing-table
```

```
*> 192.168.1.0      192.168.25.2      0
*                   192.168.35.3      0
*                   192.168.45.4      0
[R5]
```

R5 有 3 条路由，它选择了一个 router-id 较小的 R2 去往 192.168.1.0

现在要求从 R4 走

R2 R3 R4 都需要配置

R2:

```
ip ip-prefix 1 permit 192.168.1.0 24
```

```
route-policy 10 permit node 1
```

```
if-match ip-prefix 1
```

```
apply cost 200
```

```
route-policy 10 permit node 2
```

```
bgp 200
```

```
peer 192.168.25.5 route-policy 10 export
```

```
q
```


R3:

```
ip ip-prefix 1 permit 192.168.1.0 24
route-policy 10 permit node 1
if-match ip-prefix 1
apply cost 150
route-policy 10 permit node 2
bgp 200
peer 192.168.35.5 route-policy 10 export
q
```

R4:

```
ip ip-prefix 1 permit 192.168.1.0 24
route-policy 10 permit node 1
if-match ip-prefix 1
apply cost 100
route-policy 10 permit node 2
bgp 300
peer 192.168.45.5 route-policy 10 export
q
```

配置完成后在 R5 上查看，没有选择 100 的，而选择了 150 的，因为 BGP 默认只比较来自同一 AS 的 med 值

```
*> 192.168.1.0      192.168.35.3  150      0
*                  192.168.25.2  200      0
*                  192.168.45.4  100      0
[R5]
```

让 R5 强制比较来自不同 AS 且去往同一目标网络的 MED 值

R5:

```
bgp 400
compare-different-as-med
```

q

配置完成后，R5 查看 BGP 路由表

<R5>display bgp routing-table

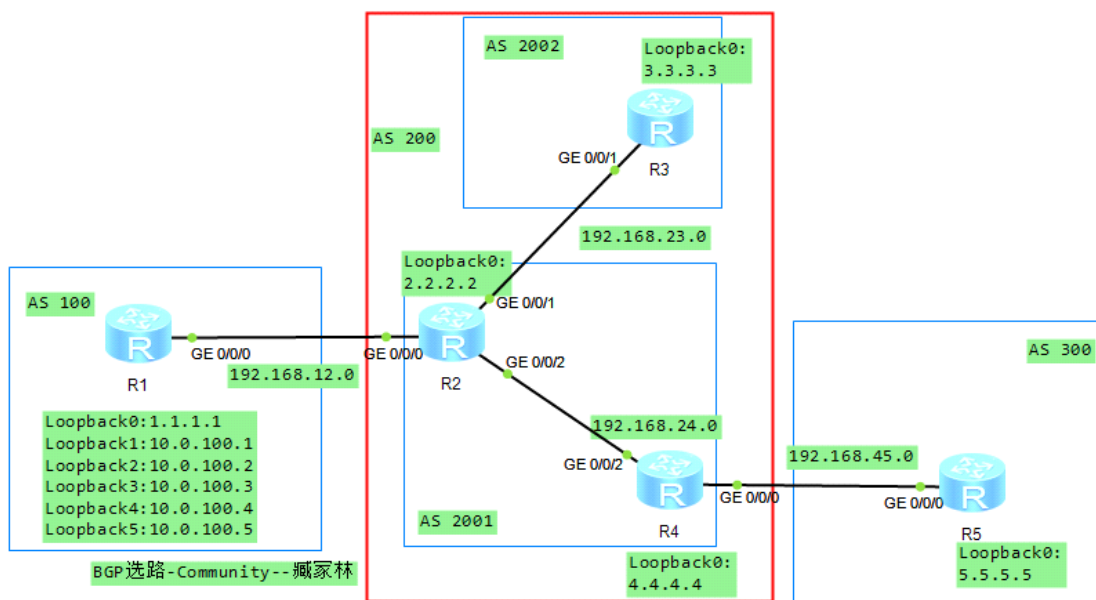
```
*> 192.168.1.0      192.168.45.4      100
*      192.168.35.3      150
*      192.168.25.2      200
```

[R5]

现在去往 192.168.1.0 是优选 R4

=====

BGP 实验 13 : BGP 选路—Community



BGP 路由的团体属性 Community 的主要作用是简化路由策略的实现过程，例如，可以将拥有团体属性的若干路由视为同一个团体，当需要对该团体中所有路由的某个特定属性进行修改时，就没必要逐一对每条路由单独进行修改。

团体属性是 BGP 路由的一种可选属性，路由器在向 BGP 对等体传递路由时，如果希望所传递的路由携带团体的属性，则需要额外的配置。

物理接口建立邻居

4 个特殊的团体：

internet ,no-export , no-advertise , no-export-Subconfed

internet : 可以向任何 BGP 对等体发布路由

no-export : 不会发给 EBGP 对等体，但可以发布给联盟 (Confederation) EBGP 对等体

no-advertise : 不会发给任何 BGP 对等体

no-export-Subconfed : 不会发给 EBGP 对等体，也不会发布给联盟 (Confederation) EBGP 对等体

全网用物理接口建立邻居，

本次的联盟范围较小，R2，R3，R4 之间可以运行 OSPF，也可以不运行

基本配置

R1:

```
undo ter mo
```

```
sys
```

```
sysname R1
```

```
user-interface console 0
```

```
idle-timeout 0 0
```

```
int loop 0
```

```
ip add 1.1.1.1 24
```

```
int loop 1
```

```
ip add 10.0.100.1 32
```

```
int loop 2
ip add 10.0.100.2 32
int loop 3
ip add 10.0.100.3 32
int loop 4
ip add 10.0.100.4 32
int loop 5
ip add 10.0.100.5 32
int g0/0/0
ip add 192.168.12.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.23.2 24
int g0/0/2
ip add 192.168.24.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
```

```
int loop 3
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.23.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/2
ip add 192.168.24.4 24
int g0/0/0
ip add 192.168.45.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/0
ip add 192.168.45.5 24
q
```

先在 AS 200 内进行 OSPF 协议的配置，这步可做，可不

R2:
ospf router-id 2.2.2.2
area 0
network 2.2.2.2 0.0.0.0
network 192.168.23.2 0.0.0.0
network 192.168.24.2 0.0.0.0
q

R3:
ospf router-id 3.3.3.3
area 0
network 3.3.3.3 0.0.0.0
network 192.168.23.3 0.0.0.0
q

R4:
ospf router-id 4.4.4.4
area 0
network 4.4.4.4 0.0.0.0
network 192.168.24.4 0.0.0.0
q

OSPF 邻居正常后，配置 BGP 协议

R2 R4 属于联盟 AS 200 成员的 AS 2001,R3 属于 AS 200 成员的 AS 2002

R1:
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
network 10.0.100.1 32
network 10.0.100.2 32
network 10.0.100.3 32
network 10.0.100.4 32

```
network 10.0.100.5 32
```

```
q
```

```
R2:
```

```
bgp 2001
```

```
router-id 2.2.2.2
```

```
confederation id 200
```

```
confederation peer-as 2002
```

```
peer 192.168.23.3 as-number 2002
```

```
peer 192.168.23.3 next-hop-local
```

```
peer 192.168.24.4 as-number 2001
```

```
peer 192.168.24.4 next-hop-local
```

```
peer 192.168.12.1 as-number 100
```

```
q
```

```
R3:
```

```
bgp 2002
```

```
router-id 3.3.3.3
```

```
confederation id 200
```

```
confederation peer-as 2001
```

```
peer 192.168.23.2 as-number 2001
```

```
q
```

```
R4:
```

```
bgp 2001
```

```
router-id 4.4.4.4
```

```
confederation id 200
```

```
peer 192.168.24.2 as-number 2001
```

```
peer 192.168.24.2 next-hop-local
```

```
peer 192.168.45.5 as-number 300
```

```
q
```

```
R5:
```

```
bgp 300
```

```
router-id 5.5.5.5
peer 192.168.45.4 as-number 200
q
```

<R2>display bgp peer BGP 邻居建立成功

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
192.168.12.1	4	100	3	2	0	00:00:27	Established
192.168.23.3	4	2002	2	4	0	00:00:08	Established
192.168.24.4	4	2001	2	4	0	00:00:00	Established

[R2]

<R5>display bgp routing-table 在 R5 上能看到 R1 的环回口路由

```
*> 10.0.100.1/32      192.168.45.4
*> 10.0.100.2/32      192.168.45.4
*> 10.0.100.3/32      192.168.45.4
*> 10.0.100.4/32      192.168.45.4
*> 10.0.100.5/32      192.168.45.4
```

[R5]

=====

使用 no-export 团体属性控制路由

10.0.100.2/32 只能被联盟 AS200 收到，不能被 AS 300 收到。

R1:

```
ip ip-prefix 2 permit 10.0.100.2 32
route-policy 10 permit node 1
if-match ip-prefix 2
apply community no-export
route-policy 10 permit node 20
```



```
bgp 100
peer 192.168.12.2 route-policy 10 export
```

默认路由器向 BGP 对等体传递路由时不会携带团体属性，需要增加配置

```
R1 :
bgp 100
peer 192.168.12.2 advertise-community
```

```
R2:
bgp 2001
peer 192.168.23.3 advertise-community
peer 192.168.24.4 advertise-community
```

```
R4:
bgp 2001
peer 192.168.45.5 advertise-community
```

在 R2 上查看携带的团体属性 [R2]dis bgp routing-table community

```
Total Number of Routes: 1
Network                NextHop          MED          LocPrf        PrefVal Community
* > 10.0.100.2/32      192.168.12.1    0
[R2]                   0               no-export
```

查看效果<R5>display bgp routing-table

```
* > 10.0.100.1/32      192.168.45.4
* > 10.0.100.3/32      192.168.45.4
* > 10.0.100.4/32      192.168.45.4
* > 10.0.100.5/32      192.168.45.4
[R5]
```

R5 上已经没有 10.0.100.2 ,在 R2 R3 R4 上有 10.0.100.2

no-export 不会发布给 EBGP 对等体，但可能发布给联盟 (Confederation) EBGP 对等体。

=====

使用 no-export-Subconfed 控制路由

10.0.100.3/32 只能被联盟 AS 200 成员 AS2001 收到，而不能被 AS2002 以及 AS 300 收到

R1:

```
ip ip-prefix 3 permit 10.0.100.3 32
route-policy 10 permit node 2
if-match ip-prefix 3
apply community no-export-subconfed
q
```

配置完成后，查看效果<R2>display bgp routing-table R2，R4 上有 10.0.100.3，R3，R5 上则没有

<R2>display bgp routing-table community

```
Total Number of Routes: 2
Network           NextHop      MED      LocPrf  PrefVal Community
*> 10.0.100.2/32   192.168.12.1 0          0        0      no-export
*> 10.0.100.3/32   192.168.12.1 0          0        0      no-export-subconfed
[R2]
```

no-export-Subconfed 路由可以在联盟成员 AS 内部传递，但不会在成员 AS 之间传递，也不会 EBGP 邻居之间传递。

=====

使用 no -advertise 控制路由

10.0.100.4/32 路由只能被 R2 收到，而不能被其它路由器收到

R1:

```
ip ip-prefix 4 permit 10.0.100.4 32
route-policy 10 permit node 3
if-match ip-prefix 4
apply community no-advertise
q
```

配置完成后，查看 <R2>display bgp routing-table

R2 上有 10.0.100.4，R3 R4 R5 上没有

<R2>display bgp routing-table community

```
Total Number of Routes: 3
Network          NextHop      MED      LocPrf    PrefVal Community
*> 10.0.100.2/32  192.168.12.1 0
*> 10.0.100.3/32  192.168.12.1 0
*> 10.0.100.4/32  192.168.12.1 0
[R2]
```

Network	NextHop	MED	LocPrf	PrefVal	Community
10.0.100.2/32	192.168.12.1	0		0	no-export
10.0.100.3/32	192.168.12.1	0		0	no-export-subconfed
10.0.100.4/32	192.168.12.1	0		0	no-advertise

路由器接收到携带 no-advertise 团体属性的路由后，不会将它发布任何 BGP 对等体。

=====

自定义团体属性控制选路

10.0.100.5 只能被 R2 R3 收到，不能被 R4 R5 收到。

正常的团体属性难以满足这样的需求，所以添加自定义团体属性

R1:

```
ip ip-prefix 5 permit 10.0.100.5 32
route-policy 10 permit node 4
if-match ip-prefix 5
```

```
apply community 100:1
q
```

在 R4 上匹配自定义团体属性 100:1

```
R4:
ip community-filter 1 permit 100:1
route-policy 10 deny node 1
if-match community-filter 1
route-policy 10 permit node 2
q
bgp 2001
peer 192.168.24.2 route-policy 10 import
q
```

R2 R3 上有 , R4 R5 上没有

<R2>display bgp routing-table community

```
Total Number of Routes: 4
Network          NextHop      MED      LocPrf  PrefVal Community
*> 10.0.100.2/32   192.168.12.1 0         0        0        no-export
*> 10.0.100.3/32   192.168.12.1 0         0        0        no-export-subconfed
*> 10.0.100.4/32   192.168.12.1 0         0        0        no-advertise
*> 10.0.100.5/32   192.168.12.1 0         0        0        <100:1>
[R2]
```

在 R1 上也可以 10.0.100.1 加上一个 internet

```
R1 :
ip ip-prefix 1 permit 10.0.100.1 32
route-policy 10 permit node 6
if-match ip-prefix 1
apply community internet
q
```

在 R2 上查看 5 个团体属性值 , 都设置好

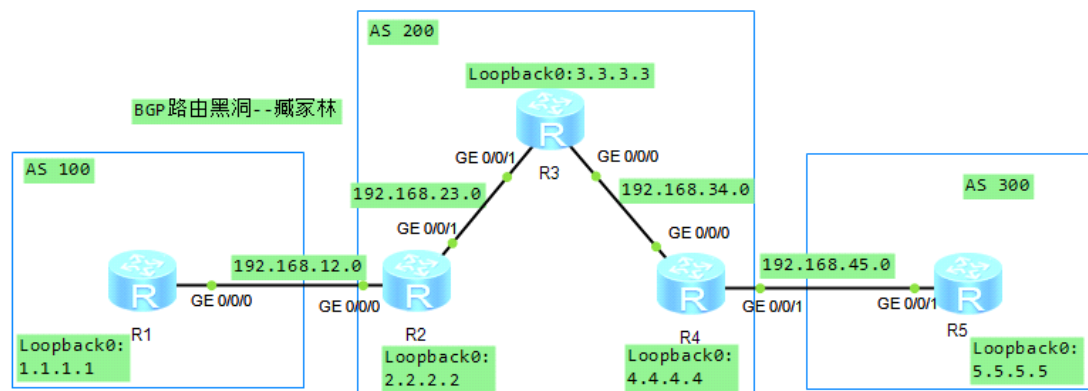
```

Total Number of Routes: 5
Network          NextHop          MED          LocPrf        PrefVal Community
* > 10.0.100.1/32 192.168.12.1      0              0             0    internet
* > 10.0.100.2/32 192.168.12.1      0              0             0    no-export
* > 10.0.100.3/32 192.168.12.1      0              0             0    no-export-subconfed
* > 10.0.100.4/32 192.168.12.1      0              0             0    no-advertise
* > 10.0.100.5/32 192.168.12.1      0              0             0    <100:1>
[R2]

```

= = = = =

BGP 实验 14 : BGP 路由黑洞



在 BGP 网络中，报文穿越 Transit AS 时，有可能会被 Transit AS 中未运行 BGP 协议的路由收到，由于这样的路由器没有 AS 间的 BGP 路由信息，报文有可能会被直接丢弃，然后路由器向报文源 IP 地址发送 ICMP Unreachable 消息，然而，由于这样的路由上没有运行 BGP 协议，很可能导致该路由器上也不存在去往报文的源 IP 地址的路由，从而使得 ICMP Unreachable 消息也无法被发送出去。如此一来，报文就无声无息的消失在这样的路由器上。这种现象被形象地称为 BGP 路由黑洞。

解决方法之一就是采用 IBGP 与 IGP 的同步机制。同步机制要

求：路由器在接收一条 IBGP 对等体发送来的路由后，必须检查自己的 IGP 路由表，只有在自己的 IGP 路由表中也存在关于这条路由的信息时，才会将该 BGP 路由发布给 EBGP 对等体。为此，可以将 BGP 协议引入 IGP 协议中，让没有运行 BGP 协议的路由器也能够获得 BGP 路由。

BGP 的第 4 条通告原则：BGP Speaker 从 IBGP 学习到的路由是否通告它的 EBGP 对等体要依 IGP 和 BGP 同步的情况来决定(华为同步默认是关闭且无法打开)。

华为默认是关闭同步的，也就是对于华为设备是不检查这一条，只要下一跳是可达的，从 IBGP 学习到的路由就会传递给它的 EBGP 对等体。

基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
```

```
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.23.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.23.3 24
int g0/0/0
ip add 192.168.34.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.34.4 24
```

```
int g0/0/1
ip add 192.168.45.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/1
ip add 192.168.45.5 24
q
```

在 AS 200 内配置 OSPF 协议

```
R2:
ospf router-id 2.2.2.2
area 0
network 2.2.2.2 0.0.0.0
network 192.168.23.2 0.0.0.0
```

```
R3:
ospf router-id 3.3.3.3
area 0
network 3.3.3.3 0.0.0.0
network 192.168.23.3 0.0.0.0
network 192.168.34.3 0.0.0.0
```

```
R4:
ospf router-id 4.4.4.4
area 0
network 4.4.4.4 0.0.0.0
```



```
network 192.168.34.4 0.0.0.0
```

在 R1 R2 R4 R5 上配置 BGP 协议，R3 先不配置

R1:

```
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
network 1.1.1.0 255.255.255.0
```

R2:

```
bgp 200
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface LoopBack 0
peer 4.4.4.4 next-hop-local
```

R4:

```
bgp 200
router-id 4.4.4.4
peer 192.168.45.5 as-number 300
peer 2.2.2.2 as-number 200
peer 2.2.2.2 connect-interface LoopBack 0
peer 2.2.2.2 next-hop-local
```

R5:

```
bgp 300
router-id 5.5.5.5
peer 192.168.45.4 as-number 200
network 5.5.5.0 255.255.255.0
```

=====

配置完成后，R2 与 R1，R2 与 R4 邻居关系已建立
R4 与 R5 邻居也建立

<R1>display bgp routing-table

<R5>display bgp routing-table

```
Total Number of Routes: 2
      Network          NextHop      MED      LocPrf    PrefVal Path/Ogn
* >  1.1.1.0/24        0.0.0.0        0
* >  5.5.5.0/24        192.168.12.2   0      200 300i
[R1]
```

R1 R5 上都有去往对方的环回口的路由，但是不通的

<R1>ping -a 1.1.1.1 5.5.5.5

[R1]ping -a 1.1.1.1 5.5.5.5

PING 5.5.5.5: 56 data bytes, press

Request time out

Request time out

Request time out

Request time out

Request time out

<R1>tracert -a 1.1.1.1 5.5.5.5

只到达了 R2，未能到达 R3

[R1]tracert -a 1.1.1.1 5.5.5.5

traceroute to 5.5.5.5(5.5.5.5), max hops
o break

1 192.168.12.2 40 ms 50 ms 50 ms

2 * * *

在 R2 查看路由表

1.1.1.0/24	EBGP	255	0	D	192.168.12.1	GigabitEthernet
2.2.2.0/24	Direct	0	0	D	2.2.2.2	LoopBack0
2.2.2.2/32	Direct	0	0	D	127.0.0.1	LoopBack0
3.3.3.3/32	OSPF	10	1	D	192.168.23.3	GigabitEthernet
4.4.4.4/32	OSPF	10	2	D	192.168.23.3	GigabitEthernet
5.5.5.0/24	IBGP	255	0	RD	4.4.4.4	GigabitEthernet

R2 去往 5.5.5.0 的路由下一跳为 4.4.4.4，递归查找，R2 去往 4.4.4.4 的下一跳为 R3。这说明 R2 上存在去往 5.5.5.0 的路由，下一跳也是可达的

在 R3 查看路由表

<R3>display ip routing-table

2.2.2.2/32	OSPF	10	1	D	192.168.23.2
3.3.3.0/24	Direct	0	0	D	3.3.3.3
3.3.3.3/32	Direct	0	0	D	127.0.0.1
4.4.4.4/32	OSPF	10	1	D	192.168.34.4
127.0.0.0/8	Direct	0	0	D	127.0.0.1
127.0.0.1/32	Direct	0	0	D	127.0.0.1
192.168.23.0/24	Direct	0	0	D	192.168.23.3

R3 没有 5.5.5.0 的路由，它会直接将报文丢弃，从 1.1.1.1 运往 5.5.5.5 的报文就在 R2 上无声无息地消失了，这就是 BGP 路由黑洞问题。

=====

R2

ospf

import-route bgp

R4:
ospf
import-route bgp

在 R3 的路由表中通过引入的方式有 1.1.1.1 和 5.5.5.5 的路由，就不会丢弃了

1.1.1.0/24	O_ASE	150	1	D	192.168.23.2
2.2.2.2/32	OSPF	10	1	D	192.168.23.2
3.3.3.0/24	Direct	0	0	D	3.3.3.3
3.3.3.3/32	Direct	0	0	D	127.0.0.1
4.4.4.4/32	OSPF	10	1	D	192.168.34.4
5.5.5.0/24	O_ASE	150	1	D	192.168.34.4

<R1>ping -a 1.1.1.1 5.5.5.5 也是通的

```
[R1]ping -a 1.1.1.1 5.5.5.5
  PING 5.5.5.5: 56 data bytes, press CTRL_C
    Reply from 5.5.5.5: bytes=56 Sequence=1 ·
    Reply from 5.5.5.5: bytes=56 Sequence=2 ·
    Reply from 5.5.5.5: bytes=56 Sequence=3 ·
    Reply from 5.5.5.5: bytes=56 Sequence=4 ·
    Reply from 5.5.5.5: bytes=56 Sequence=5 ·
```

=====

采用 IBGP 全互联方式解决 BGP 路由黑洞

R2 :
ospf
undo import-route bgp
R4 :
ospf

```
undo import-route bgp
```

R3 也运行 BGP 协议

R3:

```
bgp 200
router-id 3.3.3.3
peer 2.2.2.2 as-number 200
peer 2.2.2.2 connect-interface LoopBack 0
peer 4.4.4.4 as-number 200
peer 4.4.4.4 connect-interface LoopBack 0
```

R2:

```
bgp 200
peer 3.3.3.3 as-number 200
peer 3.3.3.3 connect-interface LoopBack 0
peer 3.3.3.3 next-hop-local
```

R4:

```
bgp 200
peer 3.3.3.3 as-number 200
peer 3.3.3.3 connect-interface LoopBack 0
peer 3.3.3.3 next-hop-local
```

<R3>display bgp peer 查看 R2 与 R2 R4 建立 BGP 邻居

<R3>display bgp routing-table

现在 R1 与 R5 的环回口就可以 ping 通了

<R1>ping -a 1.1.1.1 5.5.5.5

```
[R1]ping -a 1.1.1.1 5.5.5.5
```

```
PING 5.5.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 5.5.5.5: bytes=56 Sequence=1 ttl=252 time=140 ms
```

```
Reply from 5.5.5.5: bytes=56 Sequence=2 ttl=252 time=110 ms
```

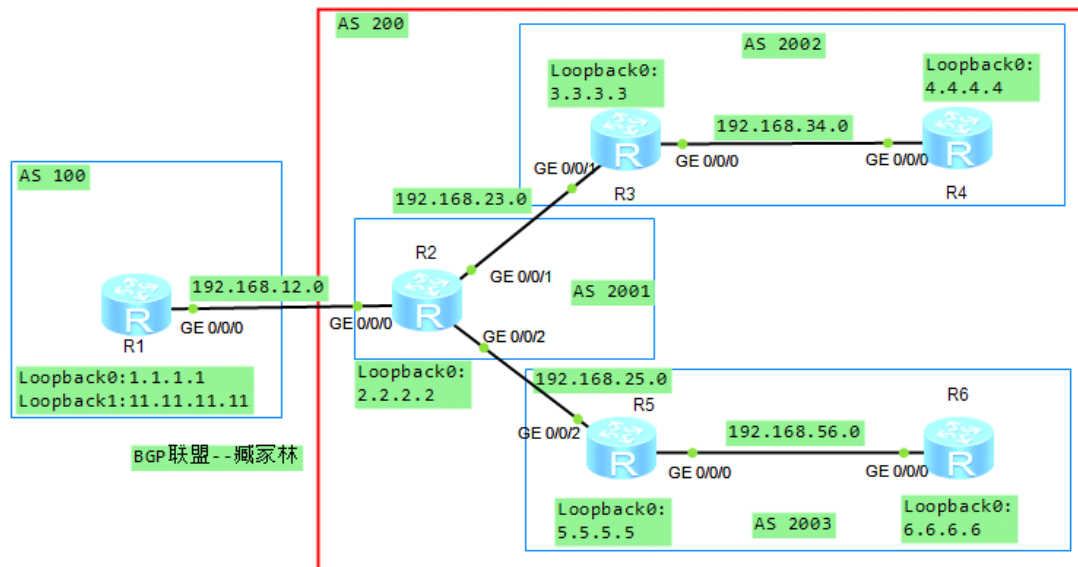
```
Reply from 5.5.5.5: bytes=56 Sequence=3 ttl=252 time=110 ms
```

```
Reply from 5.5.5.5: bytes=56 Sequence=4 ttl=252 time=100 ms
```

```
Reply from 5.5.5.5: bytes=56 Sequence=5 ttl=252 time=90 ms
```

= = = = =

BGP 实验 15 : BGP 联盟



BGP 路由反射器可以用来减少大型 AS 中 IBGP 邻居关系的数量和简化 IBGP 邻居关系的管理和维护，BGP 联盟 (Confederation) 也可以用来实现类似的目的。

一个 BGP 联盟是一个具有内部层次结构的 AS，一个 BGP 联盟由若干个子 AS 组成。对于一个 BGP 联盟，其成员 AS 内部的路由器之间需要建立全互联的 IBGP 邻居使用 BGP 路由反射器，而成员之间需要建立 EBGP 邻居关系。从联盟外的 EBGP 对等体来看，整个联盟无异于一个普通的 AS，联盟内部的结构对于联盟外的 EBGP 对等体来说是完全透明的。

物理接口建立 IBGP,EBGP 邻居

基础配置

R1:
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int loop 1
ip add 11.11.11.11 24
int g0/0/0
ip add 192.168.12.1 24
q

R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.23.2 24
int g0/0/2
ip add 192.168.25.2 24
q

R3:
undo ter mo
sys

```
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.23.3 24
int g0/0/0
ip add 192.168.34.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.34.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/2
ip add 192.168.25.5 24
int g0/0/0
```



```
ip add 192.168.56.5 24
q
```

```
R6:
undo ter mo
sys
sysname R6
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 6.6.6.6 24
int g0/0/0
ip add 192.168.56.6 24
q
```

在 AS 200 内配置 OSPF 协议

```
R2 :
ospf router-id 2.2.2.2
area 0
net 2.2.2.2 0.0.0.0
net 192.168.23.2 0.0.0.0
net 192.168.25.2 0.0.0.0
q
```

```
R3 :
ospf router-id 3.3.3.3
area 0
net 3.3.3.3 0.0.0.0
net 192.168.23.3 0.0.0.0
net 192.168.34.3 0.0.0.0
q
```

```
R4 :
```

```
ospf router-id 4.4.4.4
area 0
net 4.4.4.4 0.0.0.0
net 192.168.34.4 0.0.0.0
q
```

```
R5 :
ospf router-id 5.5.5.5
area 0
net 5.5.5.5 0.0.0.0
net 192.168.25.5 0.0.0.0
net 192.168.56.5 0.0.0.0
q
```

```
R6 :
ospf router-id 6.6.6.6
area 0
net 6.6.6.6 0.0.0.0
net 192.168.56.6 0.0.0.0
q
```

OSPF 协议正常运行之后，配置 BGP 协议

R1 R2 是 EBGp 邻居关系

```
R1 :
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 200
network 11.11.11.0 255.255.255.0
q
```

```
R2 :
bgp 2001
router-id 2.2.2.2
```

```
confederation id 200
confederation peer-as 2002 2003
peer 192.168.23.3 as-number 2002
peer 192.168.23.3 next-hop-local
peer 192.168.25.5 as-number 2003
peer 192.168.25.5 next-hop-local
peer 192.168.12.1 as-number 100
q
```

接着配置 R3 R4 R5 R6

```
R3 :
bgp 2002
router-id 3.3.3.3
confederation id 200
confederation peer-as 2001
peer 192.168.23.2 as-number 2001
peer 192.168.34.4 as-number 2002
q
```

```
R4 :
bgp 2002
router-id 4.4.4.4
confederation id 200
peer 192.168.34.3 as-number 2002
q
```

```
R5 :
bgp 2003
router-id 5.5.5.5
confederation id 200
confederation peer-as 2001
peer 192.168.25.2 as-number 2001
peer 192.168.56.6 as-number 2003
q
```

```
R6 :
bgp 2003
router-id 6.6.6.6
confederation id 200
peer 192.168.56.5 as-number 2003
q
```

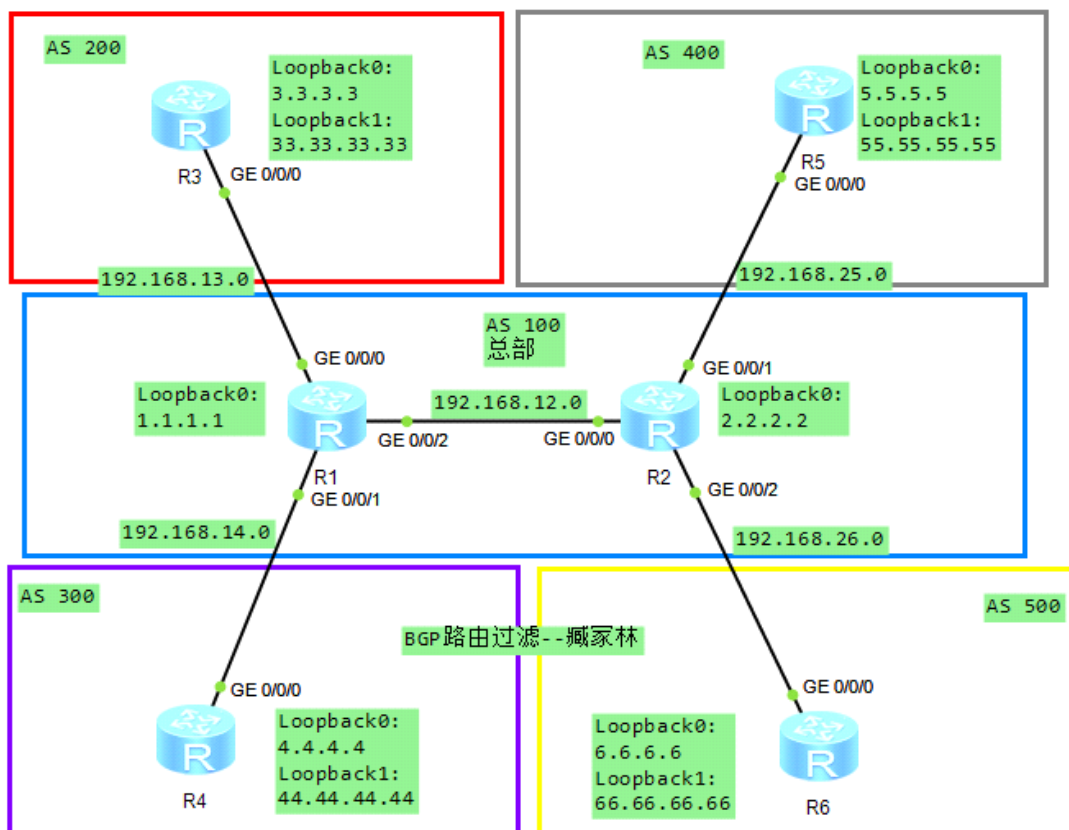
查看邻居建立情况<R2>display bgp peer
<R4>display bgp routing-table 可以看到 R1 的路由

```
Total Number of Routes: 1
      Network                NextHop          MED           LocPrf        PrefVal Path/Ogn
* >i  11.11.11.0/24          192.168.23.2      0             100           0          (2001) 100
i
[R4]
```

若公司网络规模需要扩大，则一般只需要在相应的成员 AS 中添加路由器并进行相关的配置即可，配置工作量远远小于不使用 BGP 联盟的情形。

= = = = =

BGP 实验 16 : BGP 路由过滤



路由属性的丰富性可以为实现路由过滤、路由引入等路由策略和控制提供非常有利的条件。

物理接口建立 IBGP , EBGp 邻居

基本配置

R1:

```
undo ter mo
```

```
sys
```

```
sysname R1
```

```
user-interface console 0
```

```
idle-timeout 0 0
```

```
int loop 0
```

```
ip add 1.1.1.1 24
```

```
int g0/0/0
ip add 192.168.13.1 24
int g0/0/1
ip add 192.168.14.1 24
int g0/0/2
ip add 192.168.12.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.25.2 24
int g0/0/2
ip add 192.168.26.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int loop 1
ip add 33.33.33.33 24
```

```
int g0/0/0
ip add 192.168.13.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int loop 1
ip add 44.44.44.44 24
int g0/0/0
ip add 192.168.14.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int loop 1
ip add 55.55.55.55 24
int g0/0/0
ip add 192.168.25.5 24
q
```

```
R6:
undo ter mo
```

```
sys
sysname R6
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 6.6.6.6 24
int loop 1
ip add 66.66.66.66 24
int g0/0/0
ip add 192.168.26.6 24
q
```

配置 BGP 路由协议，每台路由器都使用直连物理接口建立 BGP 邻居关系，并通告自己的环回口到 BGP 进程中。

R1:

```
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 100
peer 192.168.12.2 next-hop-local
peer 192.168.13.3 as-number 200
peer 192.168.14.4 as-number 300
network 1.1.1.0 255.255.255.0
q
```

R2:

```
bgp 100
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 192.168.12.1 next-hop-local
peer 192.168.25.5 as-number 400
peer 192.168.26.6 as-number 500
network 2.2.2.0 255.255.255.0
q
```


R3:
bgp 200
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
network 3.3.3.0 255.255.255.0
network 33.33.33.0 255.255.255.0
q

R4:
bgp 300
router-id 4.4.4.4
peer 192.168.14.1 as-number 100
network 4.4.4.0 255.255.255.0
network 44.44.44.0 255.255.255.0
q

R5:
bgp 400
router-id 5.5.5.5
peer 192.168.25.2 as-number 100
network 5.5.5.0 255.255.255.0
network 55.55.55.0 255.255.255.0
q

R6:
bgp 500
router-id 6.6.6.6
peer 192.168.26.2 as-number 100
network 6.6.6.0 255.255.255.0
network 66.66.66.0 255.255.255.0
q

查看邻居关系的建立情况<R1>display bgp peer

=====

利用 AS-Path 进行路由过滤

R3 上只允许 AS 100 的路由进入

正则表达式	涵义
^\$	匹配本地AS始发的路由
*	匹配所有路由
10	匹配所有必须通过AS10的路由
^10\$	匹配AS-PATH中只有AS10的路由
^10_	匹配从相邻AS10接收的路由
^[0-9]+\$	AS_PATH只有一个AS号

^\$

表示匹配的字符串为空，即 AS_PATH 为空，表示只匹配本地路由。

*

表示匹配任意字符串，即 AS_PATH 为任意，表示匹配所有路由。

^100

表示匹配字符串开始为 100，即 AS_PATH 最左边 AS 前 3 位（最后一个 AS）为 100、1001、1002 等，表示匹配 AS100、1001、1002 等邻居发送的路由。

^100_

表示匹配字符串开始为 100 后面为符号，即 AS_PATH 最左边 AS（最后一个 AS）为 100，表示匹配 AS100 邻居发送的路由，比较前一个表达式，“_”的好处就体现出来了，它可以和

用来帮助我们限制匹配单独的一个 AS。

`_100$`

表示匹配字符串最后为 100，即 AS_PATH 最右边 AS (起始 AS) 为 100，表示匹配 AS100 始发的路由。

`_100_`

表示字符串中间有 100，即 AS_PATH 中有 100，表示匹配经过 AS100 的路由。

`\(65535_`

表示匹配字符串为 (65535 后面为符号，即 AS_CONFED_SEQUENCE 最左边 AS (最后一个 AS) 为 65535，表示匹配联盟 AS65535 邻居发送的路由，我们知道，AS_CONFED_SEQUENCE 是用 "("、")" 表示的， "("、")" 在正则中是特殊字符，有特殊用处，所以对于这种特殊字符，可以使用 "\" 来去除其特殊意义进行匹配，同理 AS_CONFED_SET 使用的 "["、"]"，AS_SET 使用的 "{"、"}" 都可以使用 "\" 符号来去除这些特殊符号的特殊意义，举例 `\[65533_，\{202_。 \(.*_205_.*\)`

表示字符串 AS_CONFED_SEQUENCE 中间有 205，即 AS_CONFED_SEQUENCE 中有 205，表示匹配经过联盟 AS205 的路由。

`_207\)`

表示匹配字符串最后为 207)，即 AS_PATH 最右边 AS_CONFED_SEQUENCE (起始 AS) 为 207，表示匹配联盟 AS207 始发的路由。

R3 上只允许 AS 100 的路由进入
没有过滤之前，R3 上的路由有很多

```

*> 1.1.1.0/24      192.168.13.1    0      0      100i
*> 2.2.2.0/24      192.168.13.1    0      0      100i
*> 3.3.3.0/24      0.0.0.0         0      0      i
*> 4.4.4.0/24      192.168.13.1    0      0      100 300i
*> 5.5.5.0/24      192.168.13.1    0      0      100 400i
*> 6.6.6.0/24      192.168.13.1    0      0      100 500i
*> 33.33.33.0/24   0.0.0.0         0      0      i
*> 44.44.44.0/24   192.168.13.1    0      0      100 300i
*> 55.55.55.0/24   192.168.13.1    0      0      100 400i
*> 66.66.66.0/24   192.168.13.1    0      0      100 500i
[R3]

```

R3:

```

ip as-path-filter 1 permit 100$
bgp 200
peer 192.168.13.1 as-path-filter 1 import

```

查看一下效果

<R3>display bgp routing-table

R3 上只有本地路由和 AS 100 的路由

```

Total Number of Routes: 4

```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 1.1.1.0/24	192.168.13.1	0		0	100i
*> 2.2.2.0/24	192.168.13.1			0	100i
*> 3.3.3.0/24	0.0.0.0	0		0	i
*> 33.33.33.0/24	0.0.0.0	0		0	i

[R3-bgp]

=====

利用 Community 属性进行路由过滤，等一会才有效果，等待

R4 把路由只通告给 R1 R2

no-export 属性:当一台路由器，收到一个 bgp 的路由带有 no-export 属性时，这台路由器就不会再向 EBGP 对等体发送该路由了。

R4 可以将路由通告给 AS 100 中的 R1 R2 , 但不能通告给其它的路由器

<R5>display bgp routing-table 可以看到 R5 上现在是有的

```
*> 3.3.3.0/24      192.168.25.2      0      100 200i
*> 4.4.4.0/24      192.168.25.2      0      100 300i
*> 5.5.5.0/24      0.0.0.0           0      0      i
*> 6.6.6.0/24      192.168.25.2      0      100 500i
*> 33.33.33.0/24   192.168.25.2      0      100 200i
*> 44.44.44.0/24   192.168.25.2      0      100 300i
*> 55.55.55.0/24   0.0.0.0           0      0      i
*> 66.66.66.0/24   192.168.25.2      0      100 500i
[R5]
```

R4 :

```
route-policy 10 permit node 1
apply community no-export
bgp 300
peer 192.168.14.1 route-policy 10 export
peer 192.168.14.1 advertise-community
```

R1 :

```
bgp 100
peer 192.168.12.2 advertise-community
```

再去看查<R5>display bgp routing-table , 就没有了

```
*> 3.3.3.0/24      192.168.25.2      0      100 200i
*> 5.5.5.0/24      0.0.0.0           0      0      i
*> 6.6.6.0/24      192.168.25.2      0      100 500i
*> 33.33.33.0/24   192.168.25.2      0      100 200i
*> 55.55.55.0/24   0.0.0.0           0      0      i
*> 66.66.66.0/24   192.168.25.2      0      100 500i
<R5>
```

=====

利用 Next Hop 属性进行路由过滤，等一会才有效果
R6 不接收 R5 发来的路由

要求 R6 上不能接收 R5 的路由，没有配置之前，R6 上是有的

在 R2 上利用 Next Hop 属性来控制路由

R2:

```
ip ip-prefix 1 permit 192.168.25.5 32
route-policy 10 deny node 1
if-match ip next-hop ip-prefix 1
route-policy 10 permit node 2
bgp 100
peer 192.168.26.6 route-policy 10 export
```

配置好后，R2 就不向 R6 发布 5.5.5.0 和 55.55.55.0,只向 R1 发

R2 : dis bgp rout 5.5.5.0

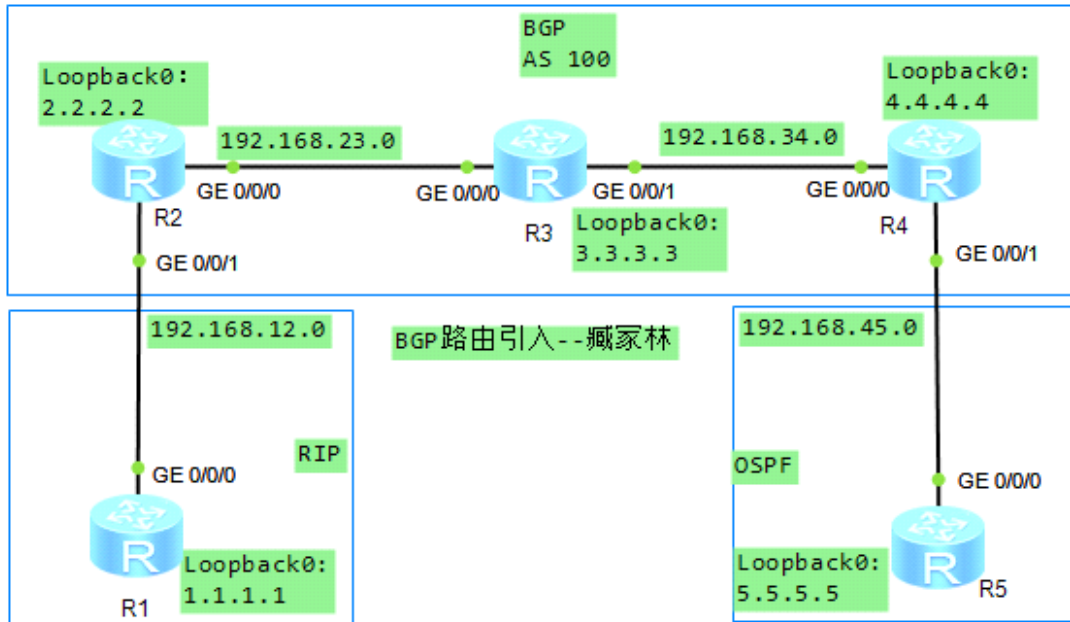
Advertised to such 1 peers:
192.168.12.1

配置完成后，在 R6 上查看一下，是没有 R5 过来的路由,要等一会才会没有 55.55.55.0 的路由
也可以 R6 refresh bgp all import

```
*> 3.3.3.0/24      192.168.26.2      0      100 200i
*> 4.4.4.0/24      192.168.26.2      0      100 300i
*> 6.6.6.0/24      0.0.0.0           0      0      i
*> 33.33.33.0/24   192.168.26.2      0      100 200i
*> 44.44.44.0/24   192.168.26.2      0      100 300i
*> 66.66.66.0/24   0.0.0.0           0      0      i
<R6>
```

= = = = =

BGP 实验 17 : BGP 路由引入



在多协议混合的网络环境中，不同的路由协议使用的协议报文各不相同，就好比说着不同的语言。如果一种路由协议需要从别的路由协议那里获取路由信息，则可以使用路由引入技术。

BGP 通过物理接口建立 IBGP 邻居，通过静态路由，实现物理网段的可达

基础配置

```
R1:  
undo ter mo  
sys  
sysname R1  
user-interface console 0  
idle-timeout 0 0
```

```
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 2.2.2.2 24
int g0/0/1
ip add 192.168.12.2 24
int g0/0/0
ip add 192.168.23.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/0
ip add 192.168.23.3 24
int g0/0/1
ip add 192.168.34.3 24
q
```



```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 4.4.4.4 24
int g0/0/0
ip add 192.168.34.4 24
int g0/0/1
ip add 192.168.45.4 24
q
```

```
R5:
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/0
ip add 192.168.45.5 24
q
```

配置 BGP 协议 R2 R3 R4

```
R2:
ip route-static 192.168.34.0 255.255.255.0
192.168.23.3
R4:
ip route-static 192.168.23.0 255.255.255.0
192.168.34.3
```

R2:

bgp 100

router-id 2.2.2.2

peer 192.168.23.3 as-number 100

peer 192.168.34.4 as-number 100

network 2.2.2.0 255.255.255.0

q

R3:

bgp 100

router-id 3.3.3.3

peer 192.168.23.2 as-number 100

peer 192.168.34.4 as-number 100

network 3.3.3.0 255.255.255.0

q

R4:

bgp 100

router-id 4.4.4.4

peer 192.168.23.2 as-number 100

peer 192.168.34.3 as-number 100

network 4.4.4.0 255.255.255.0

q

在 R1 R2 上配置 RIP 协议

R1 :

rip

version 2

network 1.0.0.0

network 192.168.12.0

q

R2 :

rip

```
version 2
network 192.168.12.0
q
```

在 R4 R5 上配置 OSPF 协议

```
R4 :
ospf router-id 4.4.4.4
area 0
network 192.168.45.4 0.0.0.0
q
```

```
R5 :
ospf router-id 5.5.5.5
area 0
network 5.5.5.5 0.0.0.0
network 192.168.45.5 0.0.0.0
q
```

现在各路由器学习到的路由是不全的

=====

引入 RIP 路由

在 R2 上将 RIP 引入到 BGP , 只引入 R1 的环回口

```
R2 :
ip ip-prefix 10 permit 1.1.1.0 24
route-policy 10 permit node 1
if-match ip-prefix 10
bgp 100
import-route rip 1 route-policy 10
```

R2 在 RIP 中下发一条默认路由

R2:

rip

default-route originate

查看效果<R2>display bgp routing-table

	Network	NextHop	MED	LocPrf	PrefVal
*>	1.1.1.0/24	0.0.0.0	1		0
*>	2.2.2.0/24	0.0.0.0	0		0
*>i	3.3.3.0/24	192.168.23.3	0	100	0
*>i	4.4.4.0/24	192.168.34.4	0	100	0
*>	192.168.12.0	0.0.0.0	0		0

[R2]

=====

引入 OSPF 路由

在 R4 上配置，将 OSPF 路由引入 BGP 中，只引入 R5 的环回口，在 OSPF 网络中环回口是 32 位的

R4:

ip ip-prefix 10 permit 5.5.5.5 32

route-policy 10 permit node 1

if-match ip-prefix 10

bgp 100

import-route ospf 1 route-policy 10

在 OSPF 中下放默认路由

R4:

ospf

default-route-advertise always

查看一下，<R4>display bgp routing-table

```

*>i 1.1.1.0/24      192.168.23.2    1      100      0
*>i 2.2.2.0/24      192.168.23.2    0      100      0
*>i 3.3.3.0/24      192.168.34.3    0      100      0
*> 4.4.4.0/24      0.0.0.0         0      0        0
*> 5.5.5.5/32      0.0.0.0         1      0        0
*>i 192.168.12.0    192.168.23.2    0      100      0
[R4]

```

R1 与 R5 之间的环回口，是可以相互 ping 通的

<R1>ping -a 1.1.1.1 5.5.5.5

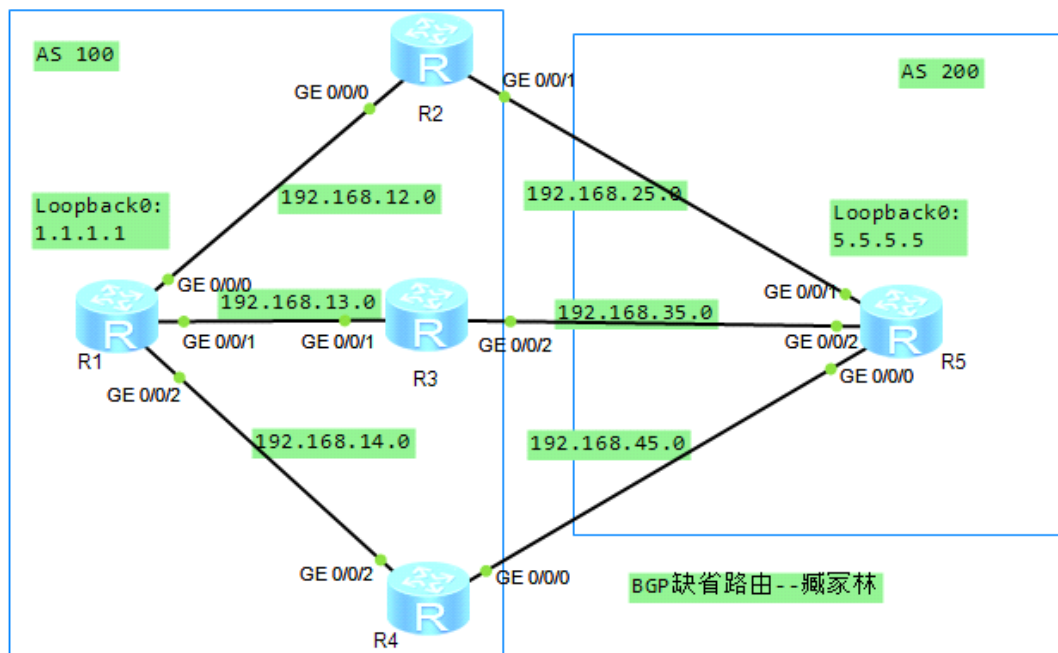
```

[R1]ping -a 1.1.1.1 5.5.5.5
PING 5.5.5.5: 56 data bytes, press CTRL_C to break
  Reply from 5.5.5.5: bytes=56 Sequence=1 ttl=252 time=180 ms
  Reply from 5.5.5.5: bytes=56 Sequence=2 ttl=252 time=130 ms
  Reply from 5.5.5.5: bytes=56 Sequence=3 ttl=252 time=140 ms
  Reply from 5.5.5.5: bytes=56 Sequence=4 ttl=252 time=130 ms
  Reply from 5.5.5.5: bytes=56 Sequence=5 ttl=252 time=170 ms

```

= = = = =

BGP 实验 18 : BGP 缺省路由



和许多其他路由协议一样，BGP 协议也支持缺省路由的使用。在实际的网络场景中，适当而灵活地运用 BGP 缺省路由，可以大大地简化繁杂的路由问题，有利于网络的优化。

在 BGP 网络中，一台路由器可以向它的一个 BGP 对等体发布一条下一跳为自己的缺省路由，也可以使用 network 向整个 AS 通告一条下一跳为自己的缺省路由，另外，还可以根据需要在 BGP 路由器上手工配置静态缺省路由。

用物理接口建立 IBGP，EBGP 邻居

基本配置

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 1.1.1.1 24
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.13.1 24
int g0/0/2
ip add 192.168.14.1 24
q
```

R2:

```
undo ter mo
sys
```

```
sysname R2
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 192.168.12.2 24
int g0/0/1
ip add 192.168.25.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int g0/0/1
ip add 192.168.13.3 24
int g0/0/2
ip add 192.168.35.3 24
q
```

```
R4:
undo ter mo
sys
sysname R4
user-interface console 0
idle-timeout 0 0
int g0/0/2
ip add 192.168.14.4 24
int g0/0/0
ip add 192.168.45.4 24
q
```

```
R5:
```

```
undo ter mo
sys
sysname R5
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 5.5.5.5 24
int g0/0/1
ip add 192.168.25.5 24
int g0/0/2
ip add 192.168.35.5 24
int g0/0/0
ip add 192.168.45.5 24
q
```

配置 BGP 路由协议，采用物理接口建立邻居。

R5 环回口通告时携带 no-advertise 团体属性，模拟出 R5 能收到 R1 的环回口，但 R1 接收不到 R5 的环回口情形。

利用默认路由的方法来解决 R1 R5 的通信。

```
R1:
bgp 100
router-id 1.1.1.1
peer 192.168.12.2 as-number 100
peer 192.168.13.3 as-number 100
peer 192.168.14.4 as-number 100
network 1.1.1.0 255.255.255.0
q
```

```
R2:
bgp 100
router-id 2.2.2.2
peer 192.168.12.1 as-number 100
peer 192.168.12.1 next-hop-local
```



```
peer 192.168.25.5 as-number 200
q
```

```
R3:
bgp 100
router-id 3.3.3.3
peer 192.168.13.1 as-number 100
peer 192.168.13.1 next-hop-local
peer 192.168.35.5 as-number 200
q
```

```
R4:
bgp 100
router-id 4.4.4.4
peer 192.168.14.1 as-number 100
peer 192.168.14.1 next-hop-local
peer 192.168.45.5 as-number 200
q
```

```
R5:
bgp 200
router-id 5.5.5.5
peer 192.168.25.2 as-number 100
peer 192.168.35.3 as-number 100
peer 192.168.45.4 as-number 100
network 5.5.5.0 255.255.255.0
q
```

配置完成后，查看路由表 R1 R5 都能学习到各自的路由

	Network	NextHop	MED	LocPrf	PrefVal
*>	1.1.1.0/24	0.0.0.0	0		0
*>i	5.5.5.0/24	192.168.12.2	0	100	0
* i		192.168.13.3	0	100	0
* i		192.168.14.4	0	100	0

[R1]

	Network	NextHop	MED	LocPrf	PrefVal
*>	1.1.1.0/24	192.168.25.2			0
*		192.168.35.3			0
*		192.168.45.4			0
*>	5.5.5.0/24	0.0.0.0	0		0

[R5]

=====

配置团体属性值，让 R1 学习不到 R5 的路由

R5 :

```

bgp 200
peer 192.168.25.2 advertise-community
peer 192.168.35.3 advertise-community
peer 192.168.45.4 advertise-community

route-policy 10 permit node 10
apply community no-advertise
bgp 200
peer 192.168.25.2 route-policy 10 export
peer 192.168.35.3 route-policy 10 export
peer 192.168.45.4 route-policy 10 export

```

查看一下，R1 收不到 R5 的环回口，但 R5 有 R1 的环回口
<R1>display bgp routing-table

```

Total Number of Routes: 1
      Network      NextHop      MED      LocPrf      PrefVal
*> 1.1.1.0/24      0.0.0.0      0
[R1]

```

R5 上有 R1 的环回口

```

      Network      NextHop      MED      LocPrf      PrefVal
*> 1.1.1.0/24      192.168.25.2
*      192.168.35.3
*      192.168.45.4
*> 5.5.5.0/24      0.0.0.0      0
[R5]

```

=====

设置默认路由

第一种方法：default-route-advertise

在 R4 上向 R1 下发一条下一跳为 R4 自己的缺省路由

R4:

bgp 100

peer 192.168.14.1 default-route-advertise

```

Total Number of Routes: 2
      Network      NextHop      MED      LocPrf      PrefVal
*>i 0.0.0.0      192.168.14.4      0      100      0
*> 1.1.1.0/24      0.0.0.0      0
[R1]

```

在 R1 上测试和 R5 环回口的还通性

<R1>tracert -a 1.1.1.1 5.5.5.5

```
[R1]tracert -a 1.1.1.1 5.5.5.5
```

```
  traceroute to 5.5.5.5(5.5.5.5), max hops: 30  
o break
```

```
 1 192.168.14.4 70 ms  60 ms  30 ms
```

```
 2 192.168.45.5 90 ms  80 ms 120 ms
```

=====

第二种方法：network 0.0.0.0 手工写一条缺省路由，在 BGP 中通告

R3:

```
ip route-static 0.0.0.0 0 NULL 0
```

```
bgp 100
```

```
network 0.0.0.0
```

Total Number of Routes: 3

Network	NextHop	MED	LocPrf	PrefVal
*>i 0.0.0.0	192.168.13.3	0	100	0
* i	192.168.14.4	0	100	0
*> 1.1.1.0/24	0.0.0.0	0		0

[R1]

=====

第三种方法：手工添加一条静态缺省路由

R1：

```
ip route-static 0.0.0.0 0 192.168.12.2
```

测试一下，R1 去往 R5 是走 R2 的

```
[R1]tracert -a 1.1.1.1 5.5.5.5
```

```
  traceroute to  5.5.5.5(5.5.5.5), max hops: 30  
o break
```

```
 1 192.168.12.2 30 ms  60 ms  70 ms
```

```
 2 192.168.25.5 100 ms 130 ms 120 ms
```

```
[R1]
```