

实验：MPLS BGP VPN 跨域

HCIE 综合实验 - MPLS BGP VPN 跨域

臧家林制作



跨域 option C 方案一 实验

ASBR 通过 MP-IBGP 向各自 AS 内的 PE 设备发布标签 IPv4 路由，并将到达本 AS 内 PE 的标签 IPv4 路由通告给它在对端 AS 的 ASBR 对等体，过渡自治系统中的 ASBR 也通告带标签的 IPv4 路由。这样，在入口 PE 和出口 PE 之间建立一条 BGP LSP。

Option C 在 Option B 的基础上进一步对 BGP 协议进行了扩展，并且将其应用在了 AS 边界，从而实现了 MBGP 部署与传统 BGPv4 部署的“完全分离”，域间 PE 之间（或者 RR 之间）建立 Multi-MPEBGP

邻居关系传递私网路由，ASBR 之间建立“普通”的 EBGP 邻居关系，ASBR 不需要同时维护 BGP IPv4 路由和 VPNv4 路由，显著降低设备性能的影响，从而更适合 VPN 数量较多的应用场景。

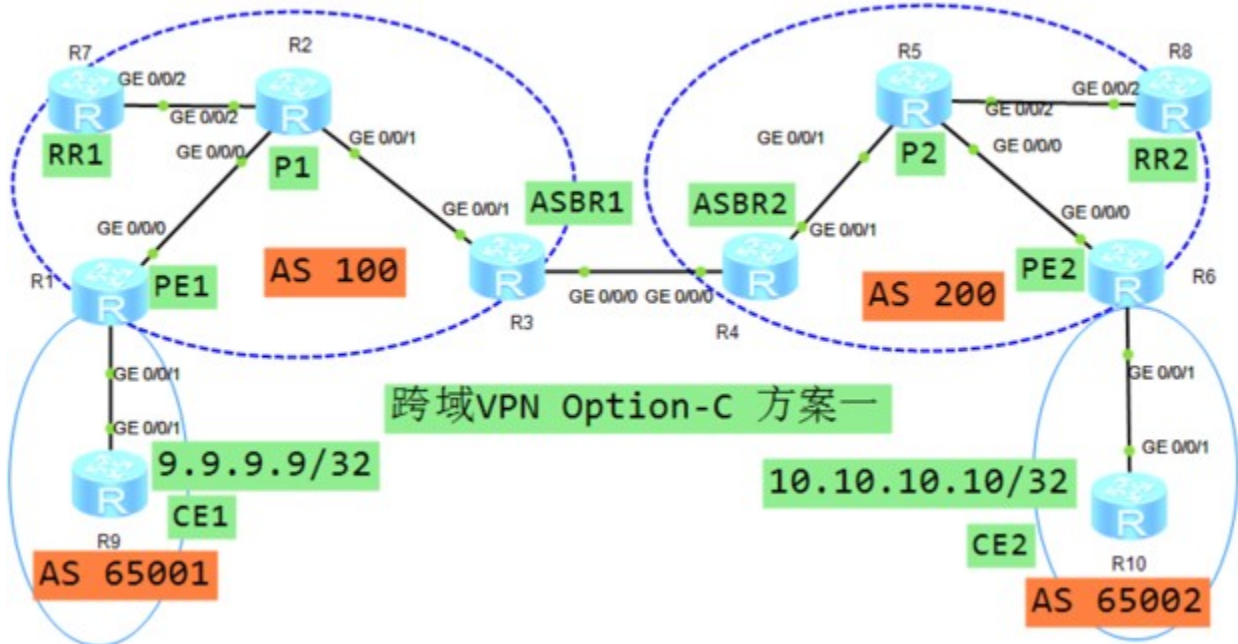
此外，由于 VPN 的路由信息只出现在 PE 设备上，而 ASBR

设备只负责报文的转发，这样就使中间域的设备可以不支持 MPLS VPN 业务，就是一个普通的支持 IP 转发路由器。在充当以上跨域的前提下，还可以同时支持普通的 IP 业务，尤其是在跨越多个域时优势更为明显，而且这个方案更适合支持 MPLS VPN 的负载分担等功能，ASBR 不会再成为网络中的性能瓶颈。不过由于这种解决方案需要对普通的 BGP 做扩展，且隧道的生成也是有别于普通的 MPLS VPN 结构，因此维护和理解起来难度较大。

不同 AS 的 PE 之间建立 Multihop 方式的 EBGp 连接，交换 VPNv4 路由。

ASBR 上不保存 VPN-IPv4 路由，相互之间也不通告 VPNv4 路由。

当网络规模较大时，可以在方案中部署 RR 设备，专门负责用户侧路由的传递。即，PE 与 RR 建立 MP-IBGP 邻居，RR1 与 RR2 建立 MP-EBGP 邻居，路由传递为 PE1-RR1-RR2-PE2，PE 之间无需直接建立 BGP 邻居关系，当 VPN 数量较多时，引入 RR 的方式可以减轻 PE 的工作负担。



配置思路

- 1.配置各接口 IP 和 OSPF 协议。
- 2.配置 MPLS/MPLS LDP。
- 3.在 PE,P,ASBR 配置与 RR 之间配置 IBGP，并设置其为 RR 的反射客户体。
- 4.在 ASBR1 与 ASBR2 之间配置 EBGP。
- 5.在 PE 上配置 VPN 实例，接口绑定 VPN。
- 6.在 PE 的 VPN 实例中配置与 CE 的 EBGP。
- 7.配置 PE 与 RR, RR 与 RR 之间的 MP-BGP。
- 8.配置 PE、RR、ASBR 交换标签 IPv4 路由能力。
- 9.配置 ASBR 发布路由应用路由策略。

=====

1.配置各接口 IP

```
R1:
undo ter mo
sy
sys R1
user-interface console 0
```

```
idle-timeout 0 0
int loo0
ip add 1.1.1.1 32
int g0/0/0
ip add 192.168.12.1 24
int g0/0/1
ip add 192.168.19.1 24
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:
undo ter mo
sy
sys R2
user-interface console 0
idle-timeout 0 0
int loo0
ip add 2.2.2.2 32
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 192.168.27.2 24
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
net 192.168.27.2 0.0.0.0
q
```

R3:

undo ter mo

sy

sys R3

user-interface console 0

idle-timeout 0 0

int loo0

ip add 3.3.3.3 32

int g0/0/0

ip add 192.168.34.3 24

int g0/0/1

ip add 192.168.23.3 24

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

R4:

undo ter mo

sy

sys R4

user-interface console 0

idle-timeout 0 0

int loo0

ip add 4.4.4.4 32

int g0/0/0

ip add 192.168.34.4 24

int g0/0/1

ip add 192.168.45.4 24

ospf router-id 4.4.4.4

area 0

net 4.4.4.4 0.0.0.0

```
net 192.168.45.4 0.0.0.0
q
```

R5:

```
undo ter mo
sy
sys R5
user-interface console 0
idle-timeout 0 0
int loo0
ip add 5.5.5.5 32
int g0/0/0
ip add 192.168.56.5 24
int g0/0/1
ip add 192.168.45.5 24
int g0/0/2
ip add 192.168.58.5 24
ospf router-id 5.5.5.5
area 0
net 5.5.5.5 0.0.0.0
net 192.168.45.5 0.0.0.0
net 192.168.56.5 0.0.0.0
net 192.168.58.5 0.0.0.0
q
```

R6:

```
undo ter mo
sy
sys R6
user-interface console 0
```

```
idle-timeout 0 0
int loo0
ip add 6.6.6.6 32
int g0/0/0
ip add 192.168.56.6 24
int g0/0/1
ip add 192.168.61.6 24
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
```

R7:

```
undo ter mo
sy
sys R7
user-interface console 0
idle-timeout 0 0
int loo0
ip add 7.7.7.7 32
int g0/0/2
ip add 192.168.27.7 24
ospf router-id 7.7.7.7
area 0
net 7.7.7.7 0.0.0.0
net 192.168.27.7 0.0.0.0
q
```

R8:

```
undo ter mo
sy
sys R8
user-interface console 0
idle-timeout 0 0
int loo0
ip add 8.8.8.8 32
int g0/0/2
ip add 192.168.58.8 24
ospf router-id 8.8.8.8
area 0
net 8.8.8.8 0.0.0.0
net 192.168.58.8 0.0.0.0
q
```

```
R9:
undo ter mo
sy
sys R9
user-interface console 0
idle-timeout 0 0
int loo0
ip add 9.9.9.9 32
int g0/0/1
ip add 192.168.19.9 24
q
```

```
R10:
undo ter mo
sy
```



```
sys R10
user-interface console 0
idle-timeout 0 0
int loo0
ip add 10.10.10.10 32
int g0/0/1
ip add 192.168.61.1 24
q
=====
```

2.配置 MPLS/MPLS LDP

```
R1 :
mpls lsr-id 1.1.1.1
mpls
mpls ldp
int g0/0/0
mpls
mpls ldp
q
```

```
R2 :
mpls lsr-id 2.2.2.2
mpls
mpls ldp
int g0/0/0
mpls
mpls ldp
int g0/0/1
mpls
mpls ldp
q
```

R3 :

mpls lsr-id 3.3.3.3

mpls

mpls ldp

int g0/0/1

mpls

mpls ldp

q

R4 :

mpls lsr-id 4.4.4.4

mpls

mpls ldp

int g0/0/1

mpls

mpls ldp

q

R5 :

mpls lsr-id 5.5.5.5

mpls

mpls ldp

int g0/0/0

mpls

mpls ldp

int g0/0/1

mpls

mpls ldp

q

```
R6 :  
mpls lsr-id 6.6.6.6  
mpls  
mpls ldp  
int g0/0/0  
mpls  
mpls ldp  
q
```

=====

3.在 PE,P,ASBR 配置与 RR 之间配置 IBGP , 并设置其为 RR 的反射客户体

```
R1:  
bgp 100  
router-id 1.1.1.1  
peer 7.7.7.7 as-n 100  
peer 7.7.7.7 con loo0  
q
```

```
R2:  
bgp 100  
router-id 2.2.2.2  
peer 7.7.7.7 as-n 100  
peer 7.7.7.7 con loo0  
q
```

```
R3:  
bgp 100
```

```
router-id 3.3.3.3
peer 7.7.7.7 as-n 100
peer 7.7.7.7 con loo0
q
```

```
R7:
bgp 100
router-id 7.7.7.7
peer 1.1.1.1 as-n 100
peer 1.1.1.1 con loo0
peer 1.1.1.1 reflect-client
peer 2.2.2.2 as-n 100
peer 2.2.2.2 con loo0
peer 2.2.2.2 reflect-client
peer 3.3.3.3 as-n 100
peer 3.3.3.3 con loo0
peer 3.3.3.3 reflect-client
q
```

=====

```
R4:
bgp 200
router-id 4.4.4.4
peer 8.8.8.8 as-n 200
peer 8.8.8.8 con loo0
q
```

```
R5:
bgp 200
```

```
router-id 5.5.5.5
peer 8.8.8.8 as-n 200
peer 8.8.8.8 con loo0
q
```

```
R6:
bgp 200
router-id 6.6.6.6
peer 8.8.8.8 as-n 200
peer 8.8.8.8 con loo0
q
```

```
R8:
bgp 200
router-id 8.8.8.8
peer 4.4.4.4 as-n 200
peer 4.4.4.4 con loo0
peer 4.4.4.4 reflect-client
peer 5.5.5.5 as-n 200
peer 5.5.5.5 con loo0
peer 5.5.5.5 reflect-client
peer 6.6.6.6 as-n 200
peer 6.6.6.6 con loo0
peer 6.6.6.6 reflect-client
q
```

=====

4.在 ASBR1 与 ASBR2 之间配置 EBGP

```
R3:
bgp 100
```

```
peer 192.168.34.4 as-n 200
q
```

```
R4:
bgp 200
peer 192.168.34.3 as-n 100
q
```

=====

5.在 PE 上配置 VPN 实例，接口绑定 VPN

```
R1:
ip vpn-instance huawei
route-distinguisher 100:1
vpn-target 100:1 both
int g0/0/1
ip binding vpn-instance huawei
ip add 192.168.19.1 24
q
```

```
R6:
ip vpn-instance huawei
route-distinguisher 100:1
vpn-target 100:1 both
int g0/0/1
ip binding vpn-instance huawei
ip add 192.168.61.6 24
q
```

=====

6.在 PE 的 VPN 实例中配置与 CE 的 EBGP

```
R1:
bgp 100
ipv4-family vpn-instance huawei
peer 192.168.19.9 as-n 65001
q
```

```
R9:
bgp 65001
router-id 9.9.9.9
peer 192.168.19.1 as-n 100
net 9.9.9.9 32
q
```

```
R6:
bgp 200
ipv4-family vpn-instance huawei
peer 192.168.61.1 as-n 65002
q
```

```
R10:
bgp 65002
router-id 10.10.10.10
peer 192.168.61.6 as-n 200
net 10.10.10.10 32
q
```

=====

7.配置 PE 与 RR, RR 与 RR 之间的 MP-BGP

peer 1.1.1.1 next-hop-invariable

配置不同 AS 域的 PE 向 EBGP 对等体发布路由时不改变下一跳；向 IBGP 对等体发布引入的 IGP 路由时使用 IGP 路由的下一跳地址。缺省情况下，BGP 发言者在向 EBGP 对等体发布路由时和向 IBGP 对等体发布引入的 IGP 路由时将下一跳改为自己的接口地址。

R1:

```
bgp 100
ipv4-family vpnv4
peer 7.7.7.7 enable
q
```

R7:

```
bgp 100
peer 8.8.8.8 as-n 200
peer 8.8.8.8 con loo0
peer 8.8.8.8 ebgp-max-hop
ipv4-family vpnv4
undo policy vpn-target
peer 1.1.1.1 enable
peer 1.1.1.1 reflect-client
peer 1.1.1.1 next-hop-invariable
peer 8.8.8.8 enable
peer 8.8.8.8 next-hop-invariable
q
```

R8:

```
bgp 200
```



```
peer 7.7.7.7 as-n 100
peer 7.7.7.7 con loo0
peer 7.7.7.7 ebgp-max-hop
ipv4-family vpnv4
undo policy vpn-target
peer 6.6.6.6 enable
peer 6.6.6.6 reflect-client
peer 6.6.6.6 next-hop-invariable
peer 7.7.7.7 enable
peer 7.7.7.7 next-hop-invariable
q
```

```
R6:
bgp 200
ipv4-family vpnv4
peer 8.8.8.8 enable
q
=====
```

8.配置 PE、RR、ASBR 交换标签 IPv4 路由能力

```
peer 7.7.7.7 label-route-capability
```

用来使能发送标签路由能力。缺省情况下，未使能发送标签路由能力。

```
R1:
bgp 100
peer 7.7.7.7 label-route-capability
q
```

R3:

bgp 100

peer 7.7.7.7 label-route-capability

peer 192.168.34.4 label-route-capability

q

R7:

bgp 100

peer 1.1.1.1 label-route-capability

peer 3.3.3.3 label-route-capability

q

=====

R4:

bgp 200

peer 8.8.8.8 label-route-capability

peer 192.168.34.3 label-route-capability

q

R6:

bgp 200

peer 8.8.8.8 label-route-capability

q

R8:

bgp 200

peer 4.4.4.4 label-route-capability

peer 6.6.6.6 label-route-capability

q

=====

9.配置 ASBR 发布路由应用路由策略

```
peer 7.7.7.7 route-policy R7 export
```

配置 ASBR1：对向 RR1 发布的路由应用路由策略，对于向本 AS 内的 RR 发布的路由，如果是带标签的 IPv4 路由，为其分配新的 MPLS 标签。

```
peer 192.168.34.4 route-policy R4 export
```

配置 ASBR1：对向 ASBR2 发布的路由应用路由策略，对于从本 AS 内的 RR 接收的路由，在向对端 AS 的 ASBR 发布时，分配 MPLS 标签。

将 PE1 和 RR1 的 Loopback 地址发布给 ASBR2，进而发布给 RR2 和 PE2。

当 BGP 的目的网络和下一跳是同一个地址时，这条路由是不可用的

R3:

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

```
mpls
```

```
q
```

```
bgp 100
```

```
network 1.1.1.1 32
```

```
network 7.7.7.7 32
```

```
q
```

```
route-policy R4 permit node 10
```

```
apply mpls-label
```

```
route-policy R7 permit node 10
```

```
if-match mpls-label
apply mpls-label
q
```

```
bgp 100
peer 7.7.7.7 route-policy R7 export
peer 192.168.34.4 route-policy R4 export
q
```

```
R4:
int g0/0/0
mpls
q
```

```
bgp 200
network 6.6.6.6 32
network 8.8.8.8 32
q
```

```
route-policy R3 permit node 10
apply mpls-label
route-policy R8 permit node 10
if-match mpls-label
apply mpls-label
q
```

```
bgp 200
peer 8.8.8.8 route-policy R8 export
peer 192.168.34.3 route-policy R3 export
q
```

=====

检查配置结果

全部配置完成后，进行查看和检测

R7:display bgp peer 查看 BGP 邻居关系状态

```
[R7]dis bgp peer
```

```
BGP local router ID : 7.7.7.7
Local AS number : 100
1.1.1.1      4      100
42           58      0 00:39:01
Established
2.2.2.2      4      100
63           77      0 01:01:03
Established
3.3.3.3      4      100
57           56      0 00:39:01
Established
8.8.8.8      4      200
8            6      0 00:00:36
Established
```

R1:display bgp vpnv4 all routing-table 查看 BGP 路由

```
[R1]dis bgp vpnv4 all routing-table
```

```
BGP Local router ID is 1.1.1.1
```

```
Total number of routes from all PE: 2
```

```
Route Distinguisher: 100:1
```

```
Network                                         NextHop
```

MED	LocPrf	PrefVal
Path/Ogn		
*> 9.9.9.9/32		
192.168.19.9	0	0
65001i		
*>i 10.10.10.10/32		
100	0	6.6.6.6
		200 65002i

VPN-Instance huawei, Router ID 1.1.1.1:

Total Number of Routes: 2

Network	NextHop
MED	PrefVal
Path/Ogn	
*> 9.9.9.9/32	
192.168.19.9	0
0	65001i
*>i 10.10.10.10/32	
100	0
	6.6.6.6
	200 65002i

R9: display bgp routing-table CE 之间能学习到对方的接口路由，能相互 ping 通

*> 9.9.9.9/32		
0.0.0.0	0	0
i		
*> 10.10.10.10/32		
		192.168.19.1

0 100 200 65002i

```
[R9]ping -a 9.9.9.9 10.10.10.10
  PING 10.10.10.10: 56      data bytes,
press CTRL_C to break
    Reply from 10.10.10.10: bytes=56
Sequence=1 ttl=250 time=140 ms
    Reply from 10.10.10.10: bytes=56
Sequence=2 ttl=250 time=160 ms
    Reply from 10.10.10.10: bytes=56
Sequence=3 ttl=250 time=160 ms
    Reply from 10.10.10.10: bytes=56
Sequence=4 ttl=250 time=210 ms
    Reply from 10.10.10.10: bytes=56
Sequence=5 ttl=250 time=170 ms
```

在 ping 的时候，在 R1 的 g0/0/0 上开启抓包，可以看到在 A
S 内部是带有 3 层标签的

1025 , 1039 , 1027

```
Frame 36: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on i
Ethernet II, Src: HuaweiTe_32:76:56 (54:89:98:32:76:56), Dst: HuaweiTe_1f:
MultiProtocol Label Switching Header, Label: 1025, Exp: 0, S: 0, TTL: 255
  0000 0000 0100 0000 0001 .... .... = MPLS Label: 1025
  .... .... .... .... 000. .... = MPLS Experimental Bits: 0
  .... .... .... .... ...0 .... = MPLS Bottom Of Label Stack: 0
  .... .... .... .... .... 1111 1111 = MPLS TTL: 255
MultiProtocol Label Switching Header, Label: 1039, Exp: 0, S: 0, TTL: 255
MultiProtocol Label Switching Header, Label: 1027, Exp: 0, S: 1, TTL: 255
Internet Protocol Version 4, Src: 9.9.9.9, Dst: 10.10.10.10
Internet Control Message Protocol
```

在 R3 的 g0/0/0 上开启抓包，可以看到在 AS 与 AS 之间是
带有 2 层标签的

1042 , 1027

```
Frame 11: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
Ethernet II, Src: HuaweiTe_39:33:06 (54:89:98:39:33:06), Dst: HuaweiTe_94:60:02 (54:89:98:94:60:02)
MultiProtocol Label Switching Header, Label: 1042, Exp: 0, S: 0, TTL: 253
  0000 0000 0100 0001 0010 .... .... = MPLS Label: 1042
  .... .... .... 000. .... = MPLS Experimental Bits: 0
  .... .... .... 0 .... = MPLS Bottom Of Label Stack: 0
  .... .... .... 1111 1101 = MPLS TTL: 253
MultiProtocol Label Switching Header, Label: 1027, Exp: 0, S: 1, TTL: 255
Internet Protocol Version 4, Src: 9.9.9.9, Dst: 10.10.10.10
Internet Control Message Protocol
```