

LAB-1

1 L2+VRRP (16分)

1.1 链路聚合

假设SW1不支持LACP，SW1和SW2互连的接口需要捆绑成一个二层逻辑接口。逻辑接口成员根据源目的MAC进行负载分担。

解法：

SW1、SW2

```
1 interface Eth-Trunk12
2     load-balance src-dst-mac
3 interface GigabitEthernet0/0/23
4     eth-trunk 12
5 interface GigabitEthernet0/0/24
6     eth-trunk 12
```

验证：

SW1、SW2

查看：According to SA-XOR-DA，还有接口 up

```
1 disp eth-trunk 12
2 Eth-Trunk12's state information is:
3 WorkingMode: NORMAL      Hash arithmetic: According to SA-XOR-DA
4 Least Active-linknumber: 1 Max Bandwidth-affected-linknumber: 8

5 Operate status: up      Number Of Up Port In Trunk: 2
6 -----
7 ---
7 PortName                Status      Weight
8 GigabitEthernet0/0/23    Up          1
9 GigabitEthernet0/0/24    Up          1
```

清除接口配置

```
1 clear configuration interface Ethernet 0/0/10
2 warning: All configurations of the interface will be cleared, and its state
3 will be shutdown. Continue? [Y/N] :y
4 -----
5 undo shutdown
```

1.2 Link-type

1. SW1,SW2,SW3,SW4互连接口的链路类型为Trunk，允许除VLAN1外的所有VLAN通过。（3）

解法：

以 SW1 为例

```

1  vlan batch 10 20
2  interface GigabitEthernet0/0/9
3      port link-type trunk
4      port trunk allow-pass vlan all
5      undo port trunk allow-pass vlan 1
6
7  interface GigabitEthernet0/0/10
8      port link-type trunk
9      port trunk allow-pass vlan all
10     undo port trunk allow-pass vlan 1
11
12 interface eth12
13     port link-type trunk
14     port trunk allow-pass vlan all
15     undo port trunk allow-pass vlan 1

```

注意：将SW3和SW4连接PC1、Server1的接口分别划分进对应VLAN

验证：

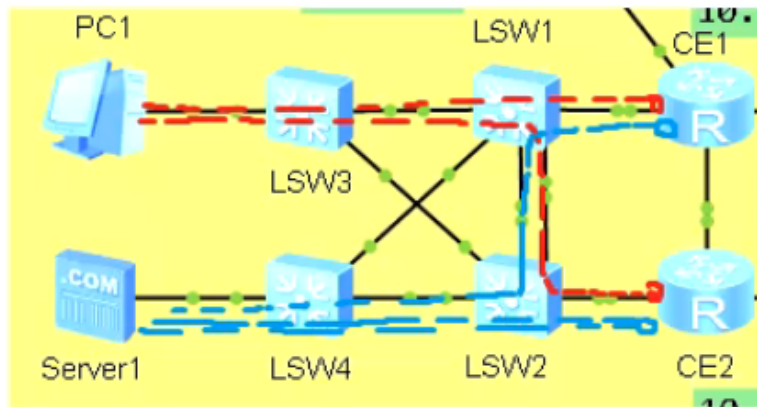
```

1  [SW1]disp port vlan active
2  T=TAG U=UNTAG
3  -----
4  Port                Link Type    PVID    VLAN List
5  -----
6  Eth-Trunk12         trunk        1        T: 10 20
7  GE0/0/1             hybrid       1        U: 1
8  GE0/0/2             hybrid       1        U: 1
9  GE0/0/3             hybrid       1        U: 1
10 GE0/0/4             hybrid       1        U: 1
11 GE0/0/5             hybrid       1        U: 1
12 GE0/0/6             hybrid       1        U: 1
13 GE0/0/7             hybrid       1        U: 1
14 GE0/0/8             hybrid       1        U: 1
15 GE0/0/9             trunk        1        T: 10 20
16 GE0/0/10            trunk        1        T: 10 20

```

- CE1, CE2的VRRP虚拟IP地址10.3.1.254, 为PC1的网关。CE1会周期性发送 sender ip 为 10.3.1.254, 源MAC为00-00-5E-00-01-01 的免费 ARP。PC1与网关之间的数据包封装在VLAN10 中 (PC1收发untag帧)
- CE1,CE2的VRRP虚拟IP地址10.3.2.254, 为Server1的网关。CE2会周期性发送sender ip为 10.3.2.254, 源MAC为00-00-5E-00-01-02 的免费 ARP。Server1与网关之间的数据包封装在 VLAN20中 (server1收发untag帧)
- VRRP的master设备重启时, 在Ge0/0/2变为UP1分钟后, 才能重新成为master. (4)

分析：



通过 CE1会周期性发送 sender ip 为10.3.1.254, 源MAC为00-00-5E-00-01-01的免费 ARP

得出 CE1 为 vlan 10 10.3.1.254 master 设备, `vrip = 1`

同理: CE2 为 vlan 20 10.3.2.254 master 设备, `vrid = 2`

解法:

注意: SW1、SW2连接CE1、CE2需要配置为trunk, CE1和CE2配置单臂路由

SW1、SW2

```
1 int g0/0/2
2   port link-type trunk
3   port trunk allow-pass all
4   undo port trunk allow-pass vlan 1
```

1. 配置单臂路由

CE1

```
1 interface GigabitEthernet0/0/2.10
2   dot1q termination vid 10
3   ip address 10.3.1.1 255.255.255.0
4   arp broadcast enable
5 interface GigabitEthernet0/0/2.20
6   dot1q termination vid 20
7   ip address 10.3.2.1 255.255.255.0
8   arp broadcast enable
```

CE2

```

1 interface GigabitEthernet0/0/2.10
2   dot1q termination vid 10
3   ip address 10.3.1.2 255.255.255.0
4   arp broadcast enable
5 interface GigabitEthernet0/0/2.20
6   dot1q termination vid 20
7   ip address 10.3.2.2 255.255.255.0
8   arp broadcast enable

```

2. 配置 MVRP

CE1

```

1 interface GigabitEthernet0/0/2.10
2   vrrp vrid 1 virtual-ip 10.3.1.254
3   vrrp vrid 1 priority 120
4 interface GigabitEthernet0/0/2.20
5   vrrp vrid 2 virtual-ip 10.3.2.254

```

CE2

```

1 interface GigabitEthernet0/0/2.10
2   ip address 10.3.1.2 255.255.255.0
3   vrrp vrid 1 virtual-ip 10.3.1.254
4 interface GigabitEthernet0/0/2.20
5   vrrp vrid 2 virtual-ip 10.3.2.254
6   vrrp vrid 2 priority 120

```

验证 CE1、CE2

```

1 [CE1]disp vrrp brief
2 Total:2      Master:1      Backup:1      Non-active:0
3 VRID  State      Interface      Type      Virtual IP
4 -----
5 1      Master      GE0/0/2.10      Normal    10.3.1.254
6 2      Backup      GE0/0/2.20      Normal    10.3.2.254
7
8 [CE2]disp vrrp brief
9 Total:2      Master:1      Backup:1      Non-active:0
10 VRID  State      Interface      Type      Virtual IP
11 -----
12 1      Backup      GE0/0/2.10      Normal    10.3.1.254
13 2      Master      GE0/0/2.20      Normal    10.3.2.254

```

3. 配置抢占延迟

CE1

```

1 int g0/0/2.10
2   vrrp vrid 1 preempt-mode timer delay 60

```

CE2

```

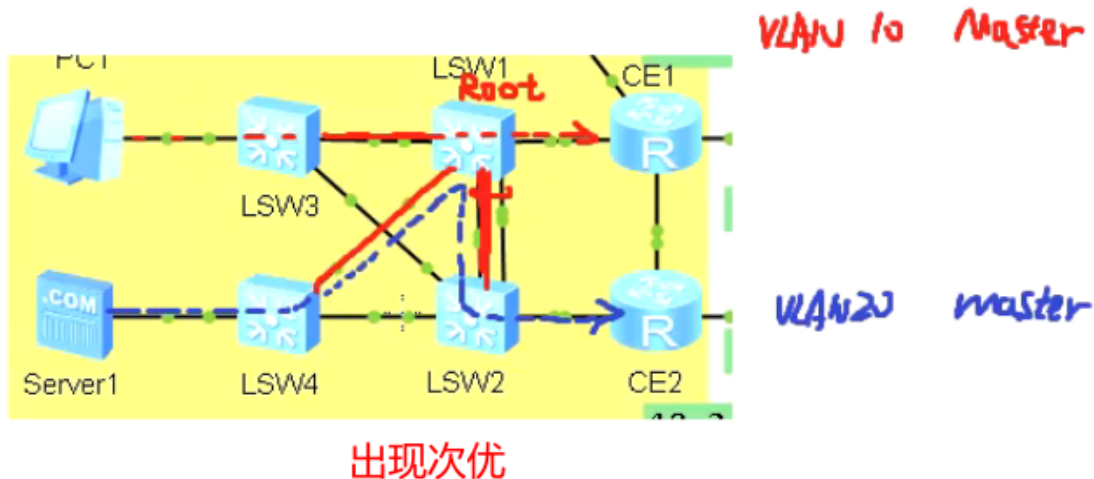
1 | int g0/0/2.20
2 | vrrp vrid 2 preempt-mode timer delay 60

```

1.3 MSTP (5分)

1. SW1,SW2,SW3,SW4都运行MSTP, VLAN10在instance10中, S1为 primary root, S2为 secondary root。VLAN20在instance20中, S1为 secondary root, S2为primary root。

分析:



解法:

SW1-SW2-SW3-SW4

```

1 | stp enable
2 | stp mode mstp
3 | stp region-configuration
4 |   region-name HUAWEI
5 |   instance 10 vlan 10
6 |   instance 20 vlan 20
7 | active region-configuration

```

SW1 配置

```

1 | stp instance 10 root primary
2 | stp instance 20 root secondary

```

SW2 配置

```

1 | stp instance 10 root secondary
2 | stp instance 20 root primary

```

2. 了交换机互连的接口, 其他接口要确保不参与MSTP计算, 立刻由disable变为forwarding状态。
(2)

分析:

配置 SW3、SW4 的边缘端口

```

1 stp bpdu-protection
2 int g0/0/1
3 stp edge-port enable

```

验证:

最好配置下 bpdu 保护, 这样可以方便查看现象, 哪个接口是 ep 端口

```

1 [SW3]disp stp instance 10 brief
2 MSTID Port Role STP State Protection
3 10 GigabitEthernet0/0/1 DESI FORWARDING BPDU
4 10 GigabitEthernet0/0/9 ALTE DISCARDING NONE
5 10 GigabitEthernet0/0/10 ROOT FORWARDING NONE
6 [SW3]disp stp instance 20 brief
7 MSTID Port Role STP State Protection
8 20 GigabitEthernet0/0/9 ROOT FORWARDING NONE
9 20 GigabitEthernet0/0/10 ALTE DISCARDING NONE

```

```

1 [SW4]disp stp instance 20 brief
2 MSTID Port Role STP State Protection
3 20 GigabitEthernet0/0/1 DESI FORWARDING NONE
4 20 GigabitEthernet0/0/9 ALTE DISCARDING NONE
5 20 GigabitEthernet0/0/10 ROOT FORWARDING NONE

```

除了 MSTID 不一样, 其余都差不多

1.4 WAN (2分)

1. PE1-RR1的互连的Serial接口, 绑定为一个逻辑接口, 成员采用HDLC。逻辑接口的IPV4,IPV6地址, 按照图1, 5配置。
2. PE3-CE3的互连接口POS接口, 绑定为一个逻辑接口, 成员采用PPP, 逻辑接口的IPV4地址, 按照图1配置。

解法:

1. PE1 - RR1 配置 ip-trunk

PE1 和 RR1 的 ser0/0/0、ser0/0/1 改为 HDLC

```

1 interface Serial0/0/0
2 link-protocol hdlc --- 输入 y 确认
3 interface Serial0/0/1
4 link-protocol hdlc

```

PE1 配置

```

1 interface Ip-Trunk1
2 ipv6 enable
3 ip address 10.1.13.1 30
4 ipv6 address 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC10/127
5 interface Serial0/0/0
6 ip-trunk 1
7 interface Serial0/0/1
8 ip-trunk 1

```

PR1 配置

```
1 interface Ip-Trunk1
2     ipv6 enable
3     ip address 10.1.13.2 30
4     ipv6 address 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC11/127
5 interface Serial0/0/0
6     ip-trunk 1
7 interface Serial0/0/1
8     ip-trunk 1
```

验证:

PE1

```
1 ping 10.1.13.2
```

```
1 [PE1]disp int Ip-Trunk 1
2 -----
3 PortName                Status      Weight
4 -----
5 Serial0/0/0              UP          1
6 Serial0/0/1              UP          1
7 -----
8 The Number of Ports in Trunk : 2
9 The Number of UP Ports in Trunk : 2
```

2. PE3 和 CE3 配置 MP-GROUP

PE3

```
1 interface Mp-group0/0/0
2     ip address 10.3.33.1 255.255.255.252
3 interface Pos5/0/0
4     ppp mp Mp-group 0/0/0
5 interface Pos6/0/0
6     ppp mp Mp-group 0/0/0
```

CE3

```
1 interface Mp-group0/0/0
2     ip address 10.3.33.2 255.255.255.252
3 interface Pos5/0/0
4     ppp mp Mp-group 0/0/0
5 interface Pos6/0/0
6     ppp mp Mp-group 0/0/0
```

验证:

```
1 [CE3]disp ppp mp int Mp-group 0/0/0
2 Mp-group is Mp-group0/0/0
3 =====Sublinks status begin=====
4 Pos5/0/0 physical UP,protocol UP
5 Pos6/0/0 physical UP,protocol UP
6 =====Sublinks status end=====
7 Physical is MP, baudrate is 310000000 bps # 注意这里带宽叠加了
```

CE3

```
1 ping 10.3.33.1
```

2 IPV4 IGP (18分)

2.1基础配置

2.2 OSPF (6分)

1. CE1和CE2之间的链路，及该两台设备的Loopback0，通告入OSPF区域。（已配）
2. CE1的Ge0/0/2.10和Ge0/0/2.20，CE2的Ge0/0/2.10和Ge0/0/2.20：直接网段通告入OSPF区域0，但这些接口不能转发OSPF报文（2）

CE1 配置

```
1 ospf 1 router-id 172.17.1.1
2 silent-interface GigabitEthernet0/0/2.10
3 silent-interface GigabitEthernet0/0/2.20
4 area 0.0.0.0
5 network 10.2.12.1 0.0.0.0
6 network 10.3.1.1 0.0.0.0
7 network 10.3.2.1 0.0.0.0
8 network 172.17.1.1 0.0.0.0
```

CE2 配置

```
1 ospf 1 router-id 172.17.1.2
2 silent-interface GigabitEthernet0/0/2.10
3 silent-interface GigabitEthernet0/0/2.20
4 area 0.0.0.0
5 network 10.2.12.2 0.0.0.0
6 network 10.3.1.2 0.0.0.0
7 network 10.3.2.2 0.0.0.0
8 network 172.17.1.2 0.0.0.0
```

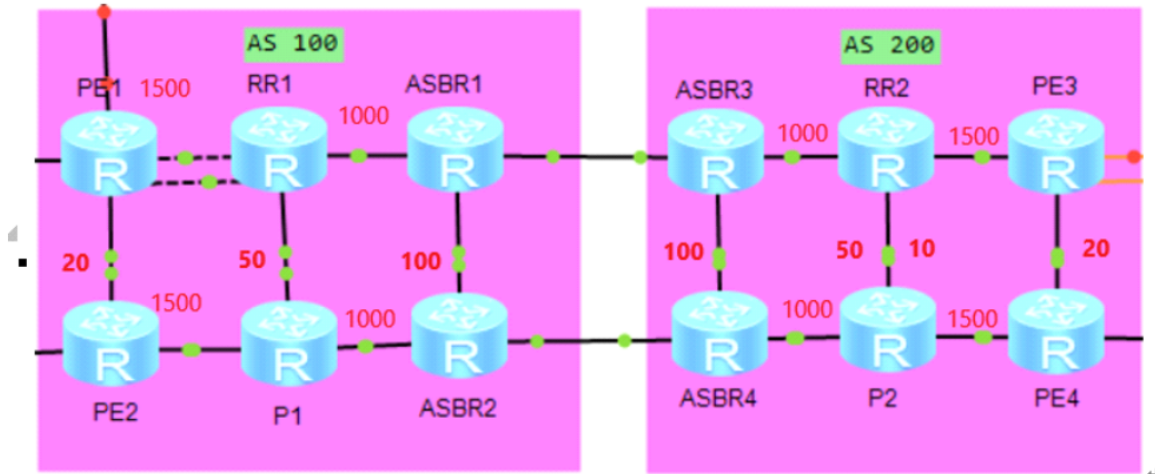
验证：disp ospf peer brief
disp ospf routing

3. RR2，P2,PE3,PE4在OSPF区域0中，cost如图2配置（已配）

建议：首先检查 router id

在系统视图下配置

IGP IPv4



1. PE3-PE4的OSPF链路类型为P2P。 (1)

```
1 int g0/0/0
2 ospf network-type p2p
```

1. PE4上将Loopback0地址引入OSPF在AS200中，各OSPF网元到PE4 Loopback0的路由要包含内部cost。 (3)

1. 匹配到 172.16.1.2/32

```
1 ip ip-prefix PE4 index 10 permit 172.16.1.2 32 greater-equal 32 less-equal 32
```

2. route-policy 标记打上 tag

```
1 route-policy D permit node 10
2 if-match ip-prefix PE4
3 apply tag 172
```

3. ospf 引入时候调用 route-policy 并设置为 type 1

```
1 ospf 1
2 import-route direct route-policy D type 1
```

检查路由开销是否累加

RR2 上查看，发现 cost = 1511 已累计

```
1 [RR2]disp ip rou 172.16.1.2
2 -----
3 Destination/Mask  Proto  Pre  Cost      Flags NextHop      Interface
4      172.16.1.2/32  O_ASE  150  1511      D    10.1.1.91.2
5      GigabitEthernet
6      0/0/0
```

2.3 ISIS

1. AS100内，loopback0和互连接口全部开启ISIS协议，其中PE1,PE2路由类型L1，区域号为49.0001，RR1，P1的路由类型L2，区域号为49.0001；ASBR1,ASBR2路由类型为L2，区域号为49.0002，各网元system-ID唯一，cost-style为wide；cost值如图2配置。（除PE1-RR1之间的逻辑链路，其余已配）（1）

解法：

```
disp cu con isis
```

```
disp cu | include isis cost
```

前面在检查一下 is-level 类型，以及 wide、cost

将PE1和RR1之间IP-trunk开启IS-IS

```
1 interface Ip-Trunk1
2 isis enable
```

检查：disp isis peer

```
1 0000.0000.0001 Ip-Trunk1 0000000002 Up 27s L1
```

2. RR2-P2的ISIS链路类型P2P。（1分）

```
1 int g0/0/0
2 isis circuit-type p2p
```

```
disp isis int g0/0/0 verbose | include p2p
```

3. 在RR2，P2上，ISIS和OSPF双向引入前缀为172.16.0.0/16的主机路由，被引入协议的cost要继承到后引入的协议中，P2和PE4的loopback0互访 走最优路径。配置要求有最好的扩展性。（8）

1. 双点双向

分析：

A、次优导致环路问题

分析：在RR2上将OSPF和ISIS执行双向引入

RR2

```
1 ospf
2 import-route isis
3 isis
4 import-route ospf
```

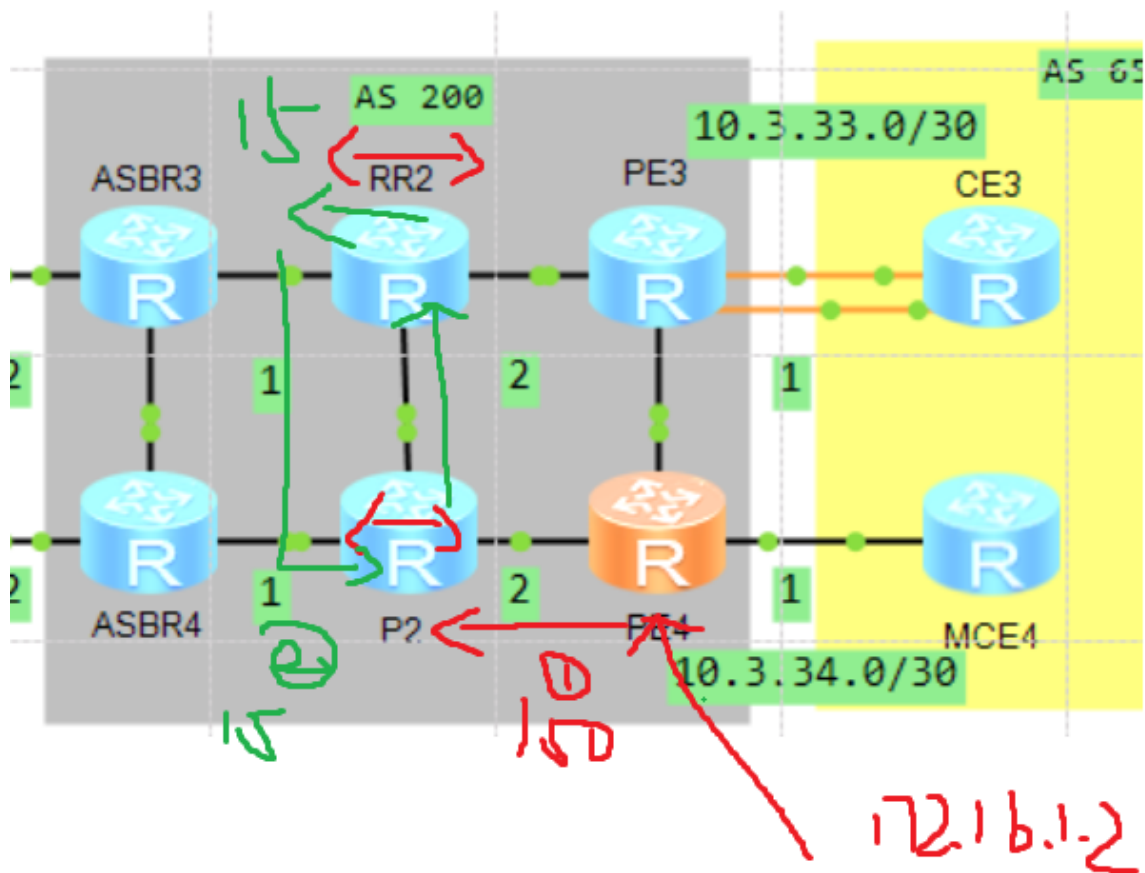
[P2]tracert -a 172.16.1.10 172.16.1.2 发现流量在P2和RR2之间出现环路

```

1 [P2]tracert -a 172.16.1.10 172.16.1.2
2
3 traceroute to 172.16.1.2(172.16.1.2), max hops: 30 ,packet length:
4 40,press CT
5 RL_C to break
6 1 10.1.91.1 20 ms 30 ms 20 ms
7 2 10.1.91.2 10 ms 10 ms 10 ms
8 ...
9 29 10.1.91.1 100 ms 90 ms 100 ms
10 30 10.1.91.2 80 ms 80 ms 90 ms

```

P2 错误优选 RR2 引入进 IS-IS 路由，导致访问 PE4 流量错误发回给 RR2，RR2 通过 OSPF 计算路由最优下一跳需要经过 P2，出现因为次优导致环路问题



解决方案：将O-ASE特定外部路由（tag 172）协议优先级修改为14

解法：

RR2、P2

```

1 route-policy PRE permit node 10
2   if-match tag 172
3   apply preference 14
4 ospf
5   preference ase route-policy PRE

```

注意：上述配置需要在RR2、P2都进行配置

B、配置要求有最好的扩展性

分析：PE4 的环回口路由失效导致 P2 无法学习到对应路由，错误优选 IS-IS 然后引入回 OSPF，都可能导致路由回灌，出现新的环路问题 此处借助 FA 地址可以防环，收到 LSA-5 发现 FA 地址为自身则不计算路由。

PE4

```
1 <PE4>reset ospf process
```

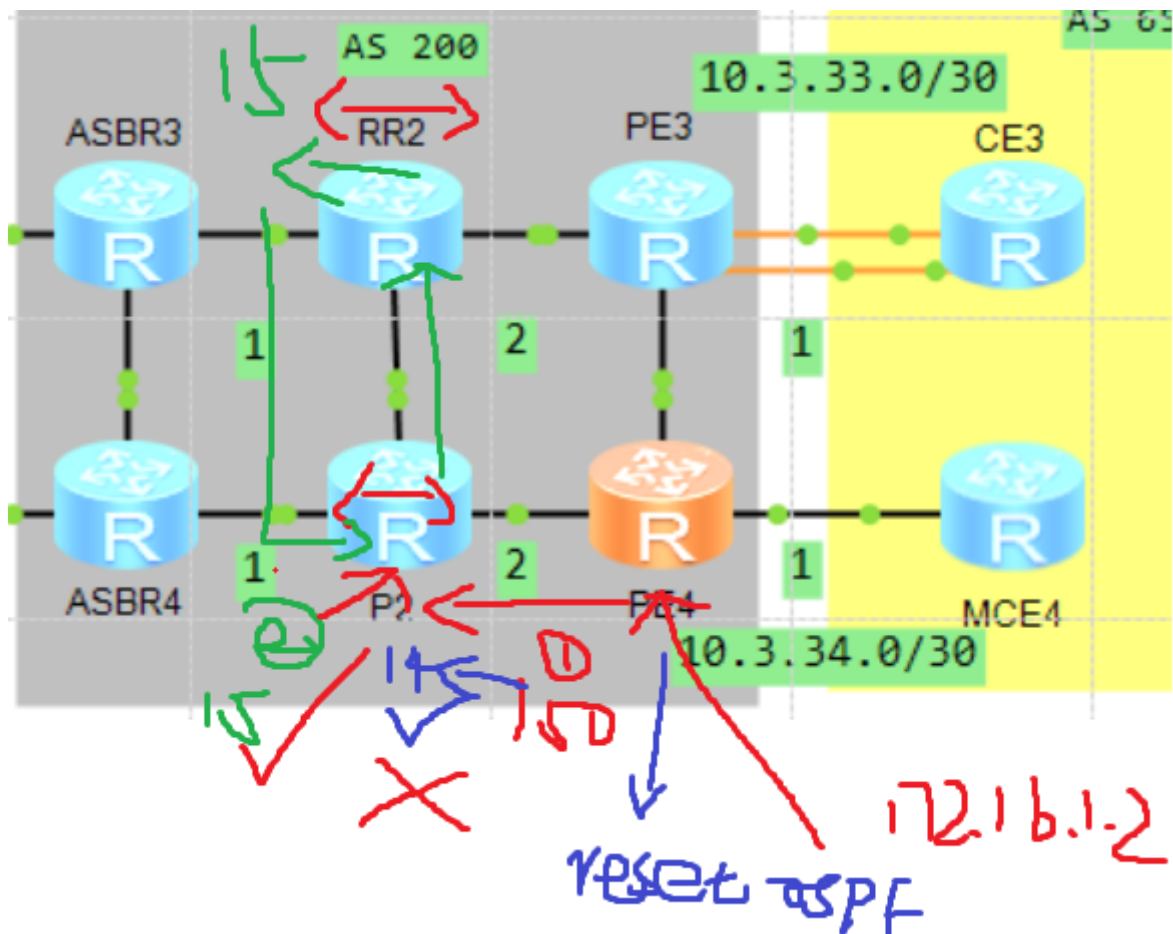
P2 马上

```
1 [P2]tracert -a 172.16.1.10 172.16.1.2
2 traceroute to 172.16.1.2(172.16.1.2), max hops: 30 ,packet length:
3 40,press CT
4 RL_C to break
5 1 10.1.102.2 20 ms 20 ms 20 ms
6 [P2]tracert -a 172.16.1.10 172.16.1.2
7 Oct 15 2021 00:38:40-08:00 P2 %%01OSPF/3/NBR_DOWN_REASON(1)[7]:Neighbor
8 state le
9 aves full or changed to Down. (ProcessId=256, NeighborRouterId=2.1.16.172,
10 Neigh
11 borAreaId=0, NeighborInterface=GigabitEthernet0/0/2,NeighborDownImmediate
12 reason
13 =Neighbor Down Due to 1-wayhello Received, NeighborDownPrimeReason=1-
14 wayhello Re
15 ceived, NeighborChangeTime=2021-10-15 00:38:40-08:00)
16 [P2]tracert -a 172.16.1.10 172.16.1.2
17 Oct 15 2021 00:38:40-08:00 P2 %%01OSPF/4/NBR_CHANGE_E(1)[8]:Neighbor changes
18 eve
19 nt: neighbor status changed. (ProcessId=256, NeighborAddress=2.102.1.10,
20 Neighbo
21 rEvent=1-Way, NeighborPreviousState=Full, NeighborCurrentState=Init)
22 [P2]tracert -a 172.16.1.10 172.16.1.2
23 traceroute to 172.16.1.2(172.16.1.2), max hops: 30 ,packet length:
24 40,press CT
25 RL_C to break
26 1 10.1.91.1 40 ms 20 ms 30 ms
27 2 10.1.91.2 10 ms 10 ms 10 ms
28 3 10.1.91.1 20 ms 20 ms 20 ms
29 4 10.1.91.2 10 ms 10 ms 20 ms
30 5 10.1.91.1 40 ms 30 ms 30 ms
31 6 10.1.91.2 20 ms 20 ms 20 ms
32 7 10.1.91.1 40 ms 30 ms 30 ms
33 8 10.1.91.2 30 ms 20 ms 30 ms
34 9 10.1.91.1 40 ms 40 ms
35 Oct 15 2021 00:38:44-08:00 P2 %%01OSPF/4/NBR_CHANGE_E(1)[9]:Neighbor changes
36 eve
37 nt: neighbor status changed. (ProcessId=256, NeighborAddress=2.102.1.10,
38 Neighbo
39 rEvent=2WayReceived, NeighborPreviousState=Init,
40 NeighborCurrentState=ExStart)
41 [P2]
42 Oct 15 2021 00:38:44-08:00 P2 %%01OSPF/4/NBR_CHANGE_E(1)[10]:Neighbor
43 changes ev
44 nt: neighbor status changed. (ProcessId=256, NeighborAddress=2.102.1.10,
45 Neighb
46 orEvent=NegotiationDone, NeighborPreviousState=ExStart,
47 NeighborCurrentState=Exc
```

```

34 hange)
35 [P2]
36 Oct 15 2021 00:38:44-08:00 P2 %%01OSPF/4/NBR_CHANGE_E(1)[11]:Neighbor
   changes ev
37 ent: neighbor status changed. (ProcessId=256, NeighborAddress=2.102.1.10,
   Neighb
38 orEvent=ExchangeDone, NeighborPreviousState=Exchange,
   NeighborCurrentState=Loadi
39 ng)
40 [P2]
41 Oct 15 2021 00:38:44-08:00 P2 %%01OSPF/4/NBR_CHANGE_E(1)[12]:Neighbor
   changes ev
42 ent: neighbor status changed. (ProcessId=256, NeighborAddress=2.102.1.10,
   Neighb
43 orEvent=LoadingDone, NeighborPreviousState=Loading,
   NeighborCurrentState=Full)
44 [P2] *
45 10 10.1.102.2 30 ms 20 ms 10 ms

```



(如果将RR2和P2之间修改为P2P则可能出现环路问题) 归根结底造成环路问题的原因是路由错误回灌, 如何防止路由回灌?

1. 双 tag 过滤

RR2

```

1 ip ip-prefix 172 permit 172.16.0.0 16 g 32 l 32
2
3 route-policy O2I deny node 10
4   if-match tag 10
5 route-policy O2I permit node 20
6   if-match ip-prefix 172
7   apply tag 20
8
9 route-policy I2O deny node 10
10  if-match tag 30
11 route-policy I2O permit node 20
12  if-match ip-prefix 172
13  apply tag 40

```

P2

```

1 ip ip-prefix 172 permit 172.16.0.0 16 g 32 l 32
2
3 route-policy O2I deny node 10
4   if-match tag 40
5 route-policy O2I permit node 20
6   if-match ip-prefix 172
7   apply tag 30
8
9 route-policy I2O deny node 10
10  if-match tag 20
11 route-policy I2O permit node 20
12  if-match ip-prefix 172
13  apply tag 10

```

2. 策略调用

RR2、PE2

```

1 ospf
2   import isis route-policy I2O
3 isis
4   import ospf route-policy O2I

```

3. 验证

查看 route-policy 中是否有 deny 掉的路由

```

1 [RR2]disp route-policy
2 Route-policy : O2I
3   deny : 10 (matched counts: 0)
4   Match clauses :
5     if-match tag 10
6   permit : 20 (matched counts: 8)
7   Match clauses :
8     if-match ip-prefix 172
9   Apply clauses :
10    apply tag 20
11 Route-policy : I2O
12   deny : 10 (matched counts: 7) # 此处有 deny
13   Match clauses :

```

```

14     if-match tag 30
15     permit : 20 (matched counts: 10)
16     Match clauses :
17         if-match ip-prefix 172
18     Apply clauses :
19     apply tag 40

```

```

1  [P2]disp route-policy
2  Route-policy : PRE
3    permit : 10 (matched counts: 40)
4    Match clauses :
5        if-match tag 172
6    Apply clauses :
7        apply preference 14
8  Route-policy : O2I
9    deny : 10 (matched counts: 0)
10   Match clauses :
11       if-match tag 40
12   permit : 20 (matched counts: 15)
13   Match clauses :
14       if-match ip-prefix 172
15   Apply clauses :
16       apply tag 30
17  Route-policy : I2O
18   deny : 10 (matched counts: 0)
19   Match clauses :
20       if-match tag 20
21   permit : 20 (matched counts: 21)
22   Match clauses :
23       if-match ip-prefix 172
24   Apply clauses :
25       apply tag 10

```

2. import-route 继承 cost 开销

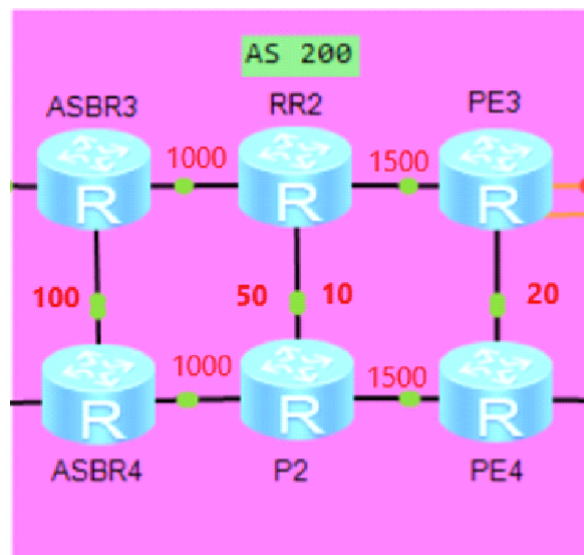
RR2、P2

```

1  ospf
2    default cost inherit-metric
3    import-route isis route-policy I2O type 1
4  isis
5    import-route ospf inherit-cost route-policy O2I

```

验证



PE4 --- ASBR3 的 cost 值 172.16.1.2/32

$1 + 1500 + 10 + 1000 = 2511$

1 是 PE4 `import-route` 进 ospf 的 cost 为 1

```

1 <ASBR3>disp ip rou 172.16.1.2
2 Route Flags: R - relay, D - download to fib
3 -----
4 Routing Table : Public
5 Summary Count : 1
6 Destination/Mask   Proto   Pre  Cost   Flags NextHop         Interface
7
8 172.16.1.2/32     ISIS-L2 15   2551   D    10.1.79.2
  GigabitEthernet0/0/1

```

3. 延迟时间

4.P1的ISIS进程：产生LSP的最大的延迟时间是1S，初始延迟为50ms，递增时间为50ms，使得LSP的快速扩散特性；SPF计算最大延迟时间是1S，初始延迟为100ms，递增时间为100ms（2）

P1 配置

```

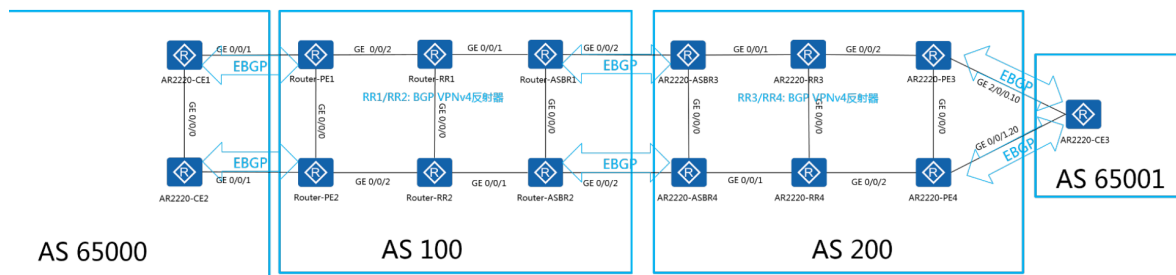
1 isis
2 timer lsp-generation 1 50 50 # 最大延迟时间 初始时间 递增时间
3 flash-flood # 快速扩散
4 timer spf 1 100 100 # 最大延迟时间 初始时间 递增时间

```

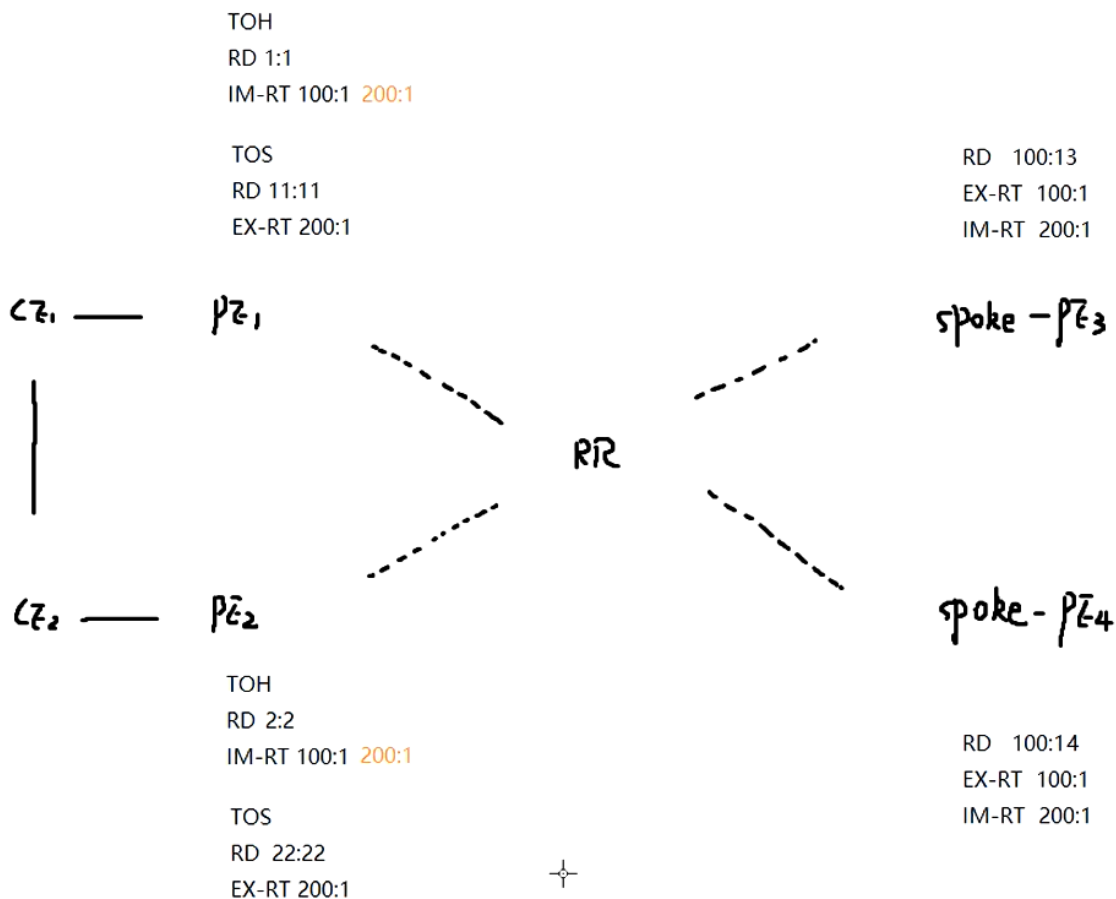
3. MPLS VPN (35分)

1. CE1,CE2为VPN1的Hub-CE，PE1,PE2为Hub-PE；CE3，CE4为VPN的spoke站点；PE3，PE4为SPOKE-PE
2. CE4为Multi-VPN-instance CE，CE4的VPN实例1，通过Ge0/0/1连接PE4。
3. 合理设置VPN1参数，使得Spoke站点互访的流量必须经过Hub-CE设备。当CE1-PE1链路断开的情况下，PE1仍然可以学习到CE1的业务路由。（PE3上的VPN1的RD为100:13,EXPORT RT为100:1，import RT为200:1）（2）

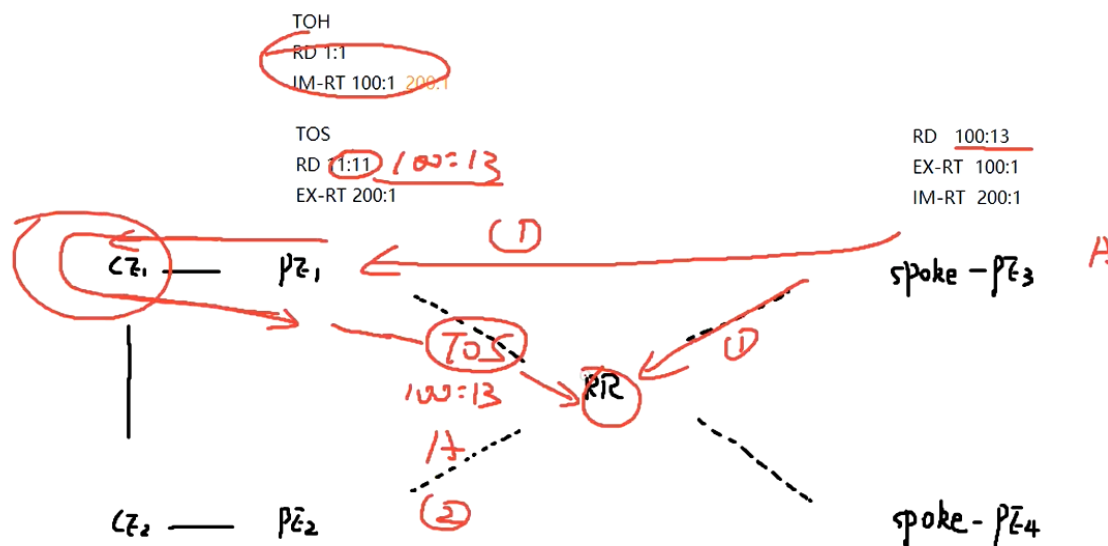
1. AS 规划图



2. RD、RT 值规划



RD 值一样 RR 会选路，次优



3. PE 创建 VPN

PE3 配置

1. 配置 RD、RT 值

```
1 ip vpn-instance VPN1
2   ipv4-family
3     route-distinguisher 100:13
4     vpn-target 100:1 export-extcommunity
5     vpn-target 200:1 import-extcommunity
```

2. 接口绑定 VPN1

```
1 [PE3]int Mp-group 0/0/0
2 [PE3-Mp-group0/0/0]disp this
3   interface Mp-group0/0/0
4     ip address 10.3.33.1 255.255.255.252
5 ip binding vpn-instance VPN1
6 ip address 10.3.33.1 255.255.255.252
```

PE4 配置

同 PE3

```
1 ip vpn-instance VPN1
2   ipv4-family
3     route-distinguisher 100:14
4     vpn-target 100:1 export-extcommunity
5     vpn-target 200:1 import-extcommunity
```

```
1 interface GigabitEthernet0/0/1
2   ip binding vpn-instance VPN1
3   ip address 10.3.34.1 255.255.255.0
```

PE1 配置

1. 配置两个 VPN，TOH --- 负责接收，TOS 负责发送

```
1 ip vpn-instance TOH
2   ipv4-family
3     route-distinguisher 1:1
4     vpn-target 100:1 200:1 import-extcommunity
5 ip vpn-instance TOS
6   ipv4-family
7     route-distinguisher 11:11
8     vpn-target 200:1 export-extcommunity
```

PE2 配置

```

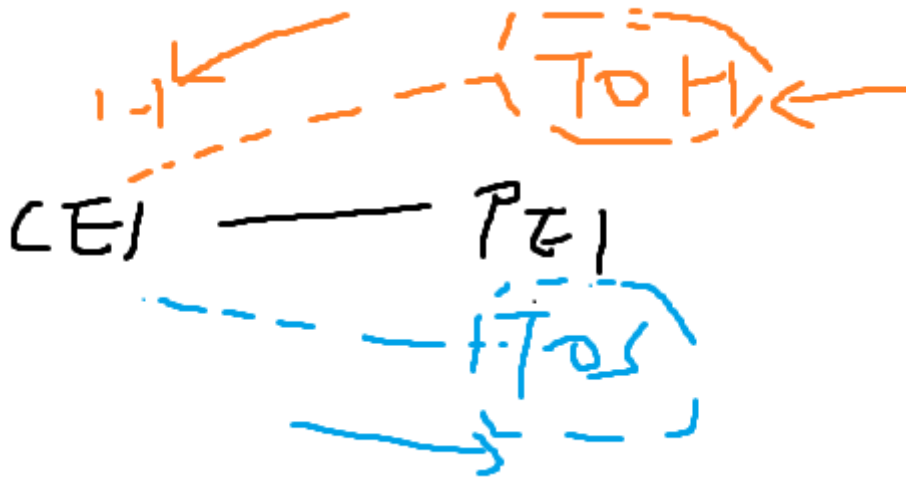
1 ip vpn-instance TOH
2   ipv4-family
3     route-distinguisher 2:2
4     vpn-target 100:1 200:1 import-extcommunity
5 ip vpn-instance TOS
6   ipv4-family
7     route-distinguisher 22:22
8     vpn-target 200:1 export-extcommunity

```

4. CE1通过G0/0/1.1和G0/0/1.2建立直接EBGP邻居，接入PE1。CE1通过G0/0/1.2,向PE1通告BGP update中，某些路由信息的ASpath中有200。在CE1上，将OSPF路由导入BGP。（2）
5. CE2通过G0/0/1.1和G0/0/1.2建立直接EBGP邻居，接入PE2。CE2通过G0/0/1.2,向PE2通告BGP update中，某些路由信息的AS-path中有 200。在CE2上，将OSPF导入BGP。（2）

分析：

也就是约定了 CE1 哪个子接口绑定哪个 VPN 实例



4. CE --- PE 配置 VPNv4 邻居

解法：

IP

设备	网段	编号
CE1	10.2.11.0/30	.2 .6
PE1	10.2.11.0/30	.1 .5
CE2	10.2.22.0/30	.2 .6
PE2	10.2.22.0/30	.1 .5

CE1 配置

```

1 interface GigabitEthernet0/0/1.1
2   ip address 10.2.11.2 255.255.255.252
3 interface GigabitEthernet0/0/1.2
4   ip address 10.2.11.6 255.255.255.252
5 bgp 65000
6   peer 10.2.11.1 as-number 100
7   peer 10.2.11.5 as-number 100
8   import ospf 1

```

CE2 配置

```

1 interface GigabitEthernet0/0/1.1
2   ip address 10.2.22.2 255.255.255.252
3 interface GigabitEthernet0/0/1.2
4   ip address 10.2.22.6 255.255.255.252
5 bgp 65000
6   peer 10.2.22.1 as-number 100
7   peer 10.2.22.5 as-number 100
8   import ospf 1

```

PE1 配置

两个子接口 分别维护两个 虚拟路由

基于两个 vpn 去建立对等体

```

1 interface GigabitEthernet0/0/1.1
2   ip binding vpn-instance TOH
3   ip address 10.2.11.1 255.255.255.252
4 interface GigabitEthernet0/0/1.2
5   ip binding vpn-instance TOS
6   ip address 10.2.11.5 255.255.255.252
7
8 bgp 100
9   ipv4-family vpn-instance TOH
10     peer 10.2.11.2 as-number 65000
11
12   ipv4-family vpn-instance TOS
13     peer 10.2.11.6 as-number 65000

```

PE2 配置

```

1 interface GigabitEthernet0/0/1.1
2   ip binding vpn-instance TOH
3   ip address 10.2.22.1 255.255.255.252
4 interface GigabitEthernet0/0/1.2
5   ip binding vpn-instance TOS
6   ip address 10.2.22.5 255.255.255.252
7
8 bgp 100
9   ipv4-family vpn-instance TOH
10     peer 10.2.22.2 as-number 65000
11
12   ipv4-family vpn-instance TOS
13     peer 10.2.22.6 as-number 65000

```

验证

PE1、PE2:

```
1 [PE1]disp bgp vpnv4 all peer
2
3 BGP local router ID : 10.1.13.1
4 Peer of IPv4-family for vpn instance :
5 VPN-Instance TOH, Router ID 10.1.13.1:
6 10.2.11.2      4      65000      5      5      0 00:03:54 Established
7 0
8 VPN-Instance TOS, Router ID 10.1.13.1:
9 10.2.11.6      4      65000      3      3      0 00:01:20 Established
10 0
```

VPNv4 路由

```
disp bgp vpnv4 all routing-table
```

```
1 *> 10.2.12.0/30      10.2.11.2      0      0      65000?
2 *> 10.3.1.0/24       10.2.11.2      0      0      65000?
3 *> 10.3.2.0/24       10.2.11.2      0      0      65000?
4 *> 172.17.1.1/32     10.2.11.2      0      0      65000?
5 *> 172.17.1.2/32     10.2.11.2      1
```

6. CE3通过OSPF区域1接入PE3，通过PE3-CE3的逻辑接口互通，通告CE3的各环回口；CE4通过OSPF区域0接入PE4，通过PE4-CE4的Ge0/0/1 接口互通，通告 CE4的各环回口；（2）

解法：

注意在之前配置 vpn 时候接口已经绑定的 vpn instance 了。

PE3 配置

```
1 ospf 10 vpn-instance VPN1
2 area 1
3 network 10.3.33.1 0.0.0.0
4 # Mp-group0/0/0
```

CE3 配置

```
1 ospf 10
2 area 1
3 network 10.3.33.2 0.0.0.0
4 network 172.17.1.3 0.0.0.0
5 network 10.3.3.3 0.0.0.0
```

PE4 配置

```
1 ospf 10 vpn-instance VPN1
2 area 0
3 network 10.3.34.1 0.0.0.0
```

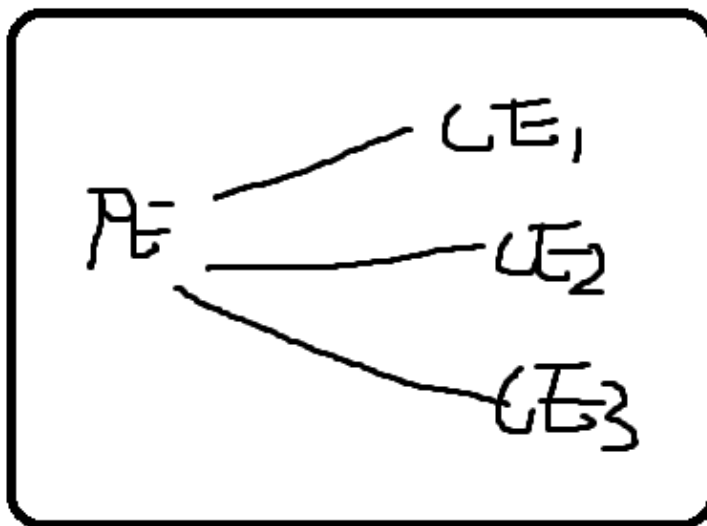
正常 CE4 配置

配置地址

```
1 interface GigabitEthernet0/0/1
2   ip address 10.3.34.2 255.255.255.0
3 interface LoopBack0
4   ip address 172.17.1.4 255.255.255.255
5 interface LoopBack1
6   ip address 10.3.4.4 255.255.255.255
```

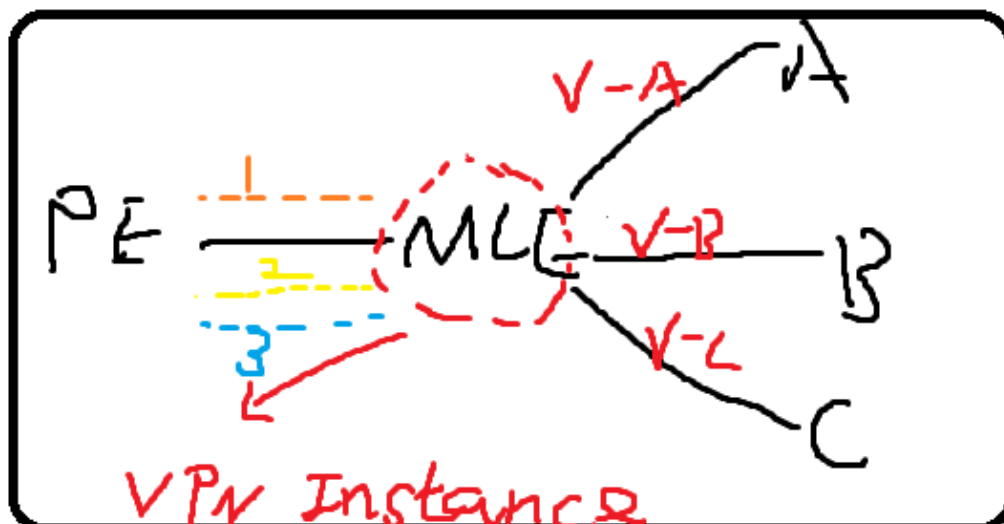
```
1 ospf 10
2   area 0
3     network 10.3.34.2 0.0.0.0
4     network 172.17.1.4 0.0.0.0
5     network 10.3.4.4 0.0.0.0
```

若是 MCE 配置



不同的 CE 对应
不同的分公司

分公司合并，把不同的
分公司放在一起办公
了，然后还有业务隔离



需要绑定实例

```
1 ip vpn-instance VPN1
2   ipv4-family
3     route-distinguisher 100:14
4 interface GigabitEthernet0/0/1
5   ip binding vpn-instance VPN1
6   ip address 10.3.34.2 255.255.255.252
7 interface LoopBack0
8   ip binding vpn-instance VPN1
9   ip address 172.17.1.4 255.255.255.255
10 interface LoopBack1
11   ip binding vpn-instance VPN1
12   ip address 10.3.4.4 255.255.255.255
```

```
1 ospf 10 vpn-instance VPN1
2   area 0
3     network 10.3.34.2 0.0.0.0
4     network 172.17.1.4 0.0.0.0
5     network 10.3.4.4 0.0.0.0
```

双 PE 防环，设置 DN 位 或者 route-tag 来防环



MCE 需关闭 DN 位

MCE 配置

```
1 ospf 10 vpn-instance VPN1
2   vpn-instance-capability simple
```

问题：

为什么有了 DN 位，还要设定 route-tag？它主要用在什么场景？

验证：检查 vpn-instance VPN1 路由

PE4

```

1 [PE4]dis ip rou vpn-instance VPN1
2 10.3.4.4/32 OSPF 10 1 D 10.3.34.2
  GigabitEthernet0/0/1
3 172.17.1.4/32 OSPF 10 1 D 10.3.34.2
  GigabitEthernet0/0/1
4 10.3.34.1/32 Direct 0 0 D 127.0.0.1
  GigabitEthernet0/0/1

```

若此时 PE4 没有学习到 CE4 上路由，请检查 ospf 邻居关系，`dis ospf error`，大概率是掩码不匹配，也就是 PE4 的接口掩码有问题。

7. 在AS100，AS200内建立IBGP IPV4邻居关系，RR1是PE1,PE2,P1,ASBR1,ASBR2的反射器，RR2是PE3，PE4，P2，ASBR4的反射器。ASBR1-ASBR3，ASBR2-ASBR4建立EBGP IPV4邻居关系。
(已预配)

8. 在ASBR上，将ISIS的loopback0路由引入BGP (2)

5. ASBR 上发布路由

解法：

ASBR1、ASBR2

```

1 ip ip-prefix 172 permit 172.16.0.0 16 greater-equal 32 less-equal 32
2 route-policy I2B permit node 10
3   if-match ip-prefix 172
4 bgp 100
5   import-route isis 1 route-policy I2B

```

ASBR3、ASBR4

```

1 ip ip-prefix 172 permit 172.16.0.0 16 greater-equal 32 less-equal 32
2 route-policy I2B permit node 10
3   if-match ip-prefix 172
4 bgp 200
5   import-route isis 1 route-policy I2B

```

检查

ASBR1 and ASBR4 上 查看是否有对端 bgp 路由

[ASBR1]dis bgp routing-table

Total Number of Routes: 14

Network	NextHop	MED	LocPrf	PrefVal	P
*> 172.16.1.1/32	0.0.0.0	1010		0	
*> 172.16.1.2/32	10.1.57.2	2551		0	
*> 172.16.1.3/32	0.0.0.0	1000		0	
*> 172.16.1.4/32	0.0.0.0	1050		0	
* i	172.16.1.6	1000	100	0	
*> 172.16.1.5/32	0.0.0.0	0		0	
* i	172.16.1.6	100	100	0	
*> 172.16.1.6/32	0.0.0.0	100		0	
*> 172.16.1.7/32	10.1.57.2	0		0	
*> 172.16.1.8/32	10.1.57.2	100		0	
*> 172.16.1.9/32	10.1.57.2	1000		0	
*> 172.16.1.10/32	10.1.57.2	1010		0	
*> 172.16.1.11/32	10.1.57.2	2500		0	
*> 172.16.1.20/32	0.0.0.0	1030		0	

[ASBR1]

[ASBR4]disp bgp routing-table

Total Number of Routes: 15

Network	NextHop	MED	LocPrf	PrefVal	P
*> 172.16.1.1/32	10.1.68.1	1060		0	
*> 172.16.1.2/32	0.0.0.0	2501		0	
*> 172.16.1.3/32	10.1.68.1	1050		0	
*> 172.16.1.4/32	10.1.68.1	1000		0	
*> 172.16.1.5/32	10.1.68.1	100		0	
*> 172.16.1.6/32	10.1.68.1	0		0	
*> 172.16.1.7/32	0.0.0.0	100		0	
*> 172.16.1.8/32	0.0.0.0	0		0	
* i	172.16.1.7	100	100	0	
*> 172.16.1.9/32	0.0.0.0	1010		0	
* i	172.16.1.7	1000	100	0	
*> 172.16.1.10/32	0.0.0.0	1000		0	
*> 172.16.1.11/32	0.0.0.0	2510		0	
* i	172.16.1.7	2500	100	0	
*> 172.16.1.20/32	10.1.68.1	1080		0	

[ASBR4]

建议：配置 ASBR 指向 RR 配置 next-hop-local

因为 LAB2 必配，不用纠结了

为什么 lab1 不用配、lab2 必配，分析

```
1 bgp 100
2 peer 172.16.1.3 next-hop-local
```

```
1 bgp 200
2 peer 172.16.1.9 next-hop-local
```

9. AS100, AS200内各网元配置MPLS LSR-ID, 全局使能MPLS,MPLS LDP (已配) AS100,AS200内各有直连链路建立LDP邻居 (除 PE1-RR1之间, 其余已配) (1)

PE1、RR1 配置

```

1 interface Ip-Trunk1
2   mpls
3   mpls ldp

```

验证:

RR1 看到有 3 条会话就 ok 了

```

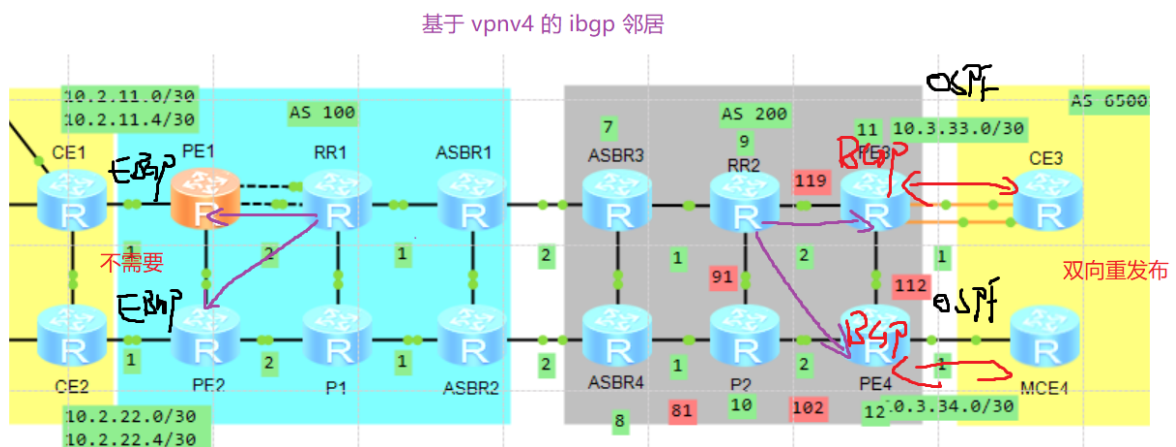
1 [RR1]disp mpls ldp session all
2 -----
3 PeerID              Status      LAM  SsnRole  SsnAge      KASent/Rcv
4 -----
5 172.16.1.1:0         Operational DU    Active  0000:00:00 1/1
6 172.16.1.4:0         Operational DU    Passive 0000:01:29 357/357
7 172.16.1.5:0         Operational DU    Passive 0000:01:29 357/357
8 -----
9 TOTAL: 3 session(s) Found.

```

6. OPTION C方案一

10. 如图4，各站点，通过MPLS BGP VPN跨域OPTION C方案一，能够相互学习路由，MPLS域不能出现次优路径。（15）

分析:



1. PE3 和 PE4 将 OSPF 和 BGP 双向引入

```

1 bgp 200
2   ipv4-family vpn-instance VPN1
3     import-route ospf 10
4   ospf 10
5     import-route bgp

```

2. AS 内部 PE 和 RR 之间 VPNv4 的对等体

PE1、PE2

```

1 bgp 100
2   ipv4-family vpnv4
3     peer 172.16.1.3 enable

```

RR1

```
1 bgp 100
2   ipv4-family vpnv4
3     policy vpn-target
4     peer 172.16.1.1 enable
5     peer 172.16.1.1 reflect-client
6     peer 172.16.1.20 enable
7     peer 172.16.1.20 reflect-client
```

RR1 关闭 RT 值检测

```
1 bgp 100
2   ipv4-family vpnv4
3     undo policy vpn-target
```

PE3、PE4

```
1 bgp 200
2   ipv4-family vpnv4
3     peer 172.16.1.9 enable
```

RR2

```
1 bgp 200
2   ipv4-family vpnv4
3     policy vpn-target
4     peer 172.16.1.11 enable
5     peer 172.16.1.11 reflect-client
6     peer 172.16.1.2 enable
7     peer 172.16.1.2 reflect-client
```

关闭 RT 值检测, RR 你只是帮忙转发一下, 真正要检测的是 PE 去做的

```
1 bgp 200
2   ipv4-family vpnv4
3     undo policy vpn-target
```

检查:

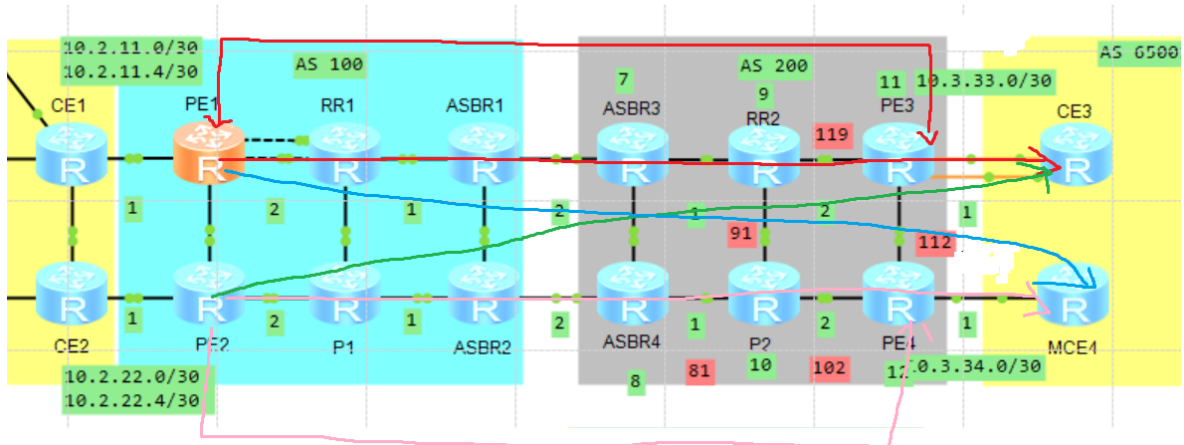
RR1、RR2: disp bgp vpnv4 all peer 、 disp bgp vpnv4 all routing-table

问题: 为什么要让 PE 把路由传给 RR 呢?

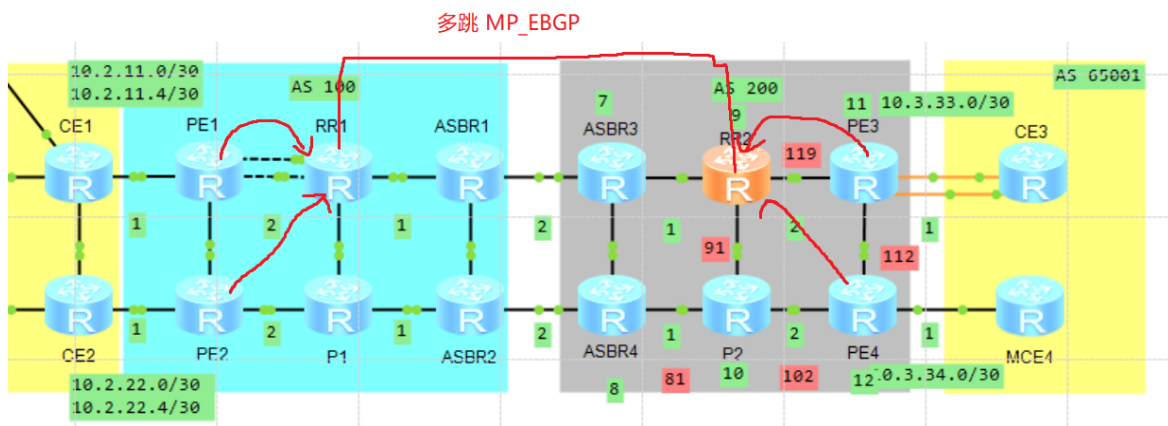
以前都是 PE 和 CE 之间互传路由, 中间的 RR、ASBR 都不需要传递 VPNv4 路由, 那是简单的跨域需求

但是现在的需求是, 需要多个站点间互访, 就不能这么干了。

此时如果有多个互访需求，则需要建立四个对等体（全互联），维护麻烦



现在的做法呢，只需要 PE 把路由传递给 RR，然后两个 RR 之间建立一个跨域的多跳 MP_EBGP

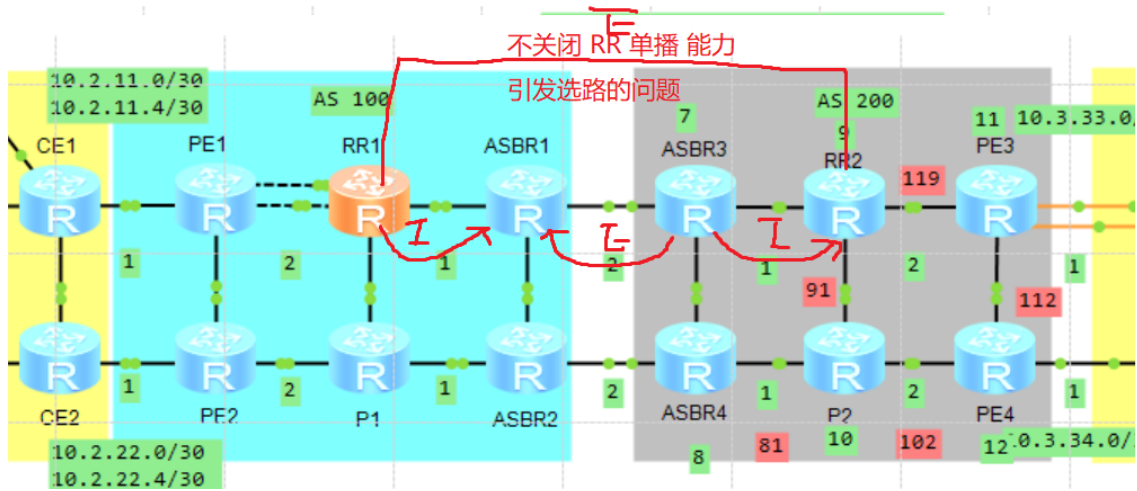


3. RR 之间建立一个跨域的多跳 MP_EBGP，传递 VPNv4 路由

RR1 配置

```
1 bgp 100
2   peer 172.16.1.9 as-number 200
3   peer 172.16.1.9 ebgp-max-hop 255
4   peer 172.16.1.9 connect-interface LoopBack0
5
6   ipv4-family vpnv4
7     peer 172.16.1.9 enable
```

建议：关闭 RR 之间的 IPv4 单播路由传递能力，只传递 VPNv4 路由



原因：查看 ASBR 表项的时候会有选路问题、查表失败

```
1 bgp 100
2   ipv4-family unicast
3     undo peer 172.16.1.9 enable
```

RR2配置

```
1 bgp 200
2   peer 172.16.1.3 as-number 100
3   peer 172.16.1.3 ebgp-max-hop 255
4   peer 172.16.1.3 connect-interface LoopBack0
5
6   ipv4-family vpnv4
7     peer 172.16.1.3 enable
```

建议：关闭 RR 之间的单播路由传递

```
1 bgp 200
2   ipv4-family unicast
3     undo peer 172.16.1.3 enable
```

检查：

RR1 RR2: disp bgp vpnv4 all peer、disp bgp vpnv4 all routing

我丢，为啥这里的邻居建立花了好多分钟 --- 其实没有

RR1 为例查看

```
*>i 172.17.1.2/32 172.16.1.20 0 100 0 65000?
Route Distinguisher: 100:13

Network NextHop MED LocPrf PrefVal Path/Ogn
*> 10.3.3.3/32 172.16.1.9 0 200?
*> 10.3.33.0/30 172.16.1.9 0 200?
*> 172.17.1.3/32 172.16.1.9 0 200?
Route Distinguisher: 100:14

Network NextHop MED LocPrf PrefVal Path/Ogn
*> 10.3.4.4/32 172.16.1.9 0 200?
*> 10.3.34.0/24 172.16.1.9 0 200?
*> 172.17.1.4/32 172.16.1.9 0 200?
[RR1-bgp]
```

4. 建立 PE 之间完整的 LSP，此处采用 方案一

分析：

A. ASBR 之间通过 BGP-Label Unicast，为自身单播分配并通过标签

ASBR1 配置

```
1 route-policy ASBR permit node 10
2   apply mpls-label
3 bgp 100
4   peer 10.1.57.2 route-policy ASBR export
5   peer 10.1.57.2 label-route-capability
```

ASBR3 配置

```
1 route-policy ASBR permit node 10
2   apply mpls-label
3 bgp 200
4   peer 10.1.57.1 route-policy ASBR export
5   peer 10.1.57.1 label-route-capability
```

ASBR2 配置

```
1 route-policy ASBR permit node 10
2   apply mpls-label
3 bgp 100
4   peer 10.1.68.2 route-policy ASBR export
5   peer 10.1.68.2 label-route-capability
```

ASBR4 配置

```
1 route-policy ASBR permit node 10
2   apply mpls-label
3 bgp 200
4   peer 10.1.68.1 route-policy ASBR export
5   peer 10.1.68.1 label-route-capability
```

检查:

ASBR1

```
1 [ASBR1]disp bgp routing-table label
2 Total Number of Routes: 14          分配/学习
3      Network      NextHop      In/Out Label
4 * > 172.16.1.1    10.1.35.1    1030/NULL
5 * > 172.16.1.2    10.1.57.2    NULL/1034
6 * > 172.16.1.3    10.1.35.1    1029/NULL
7 * > 172.16.1.4    10.1.35.1    1032/NULL
8 * i              172.16.1.6
9 * > 172.16.1.5    127.0.0.1    1034/NULL
10 * i             172.16.1.6
11 * > 172.16.1.6    10.1.56.2    1033/NULL
12 * > 172.16.1.7    10.1.57.2    NULL/1035
13 * > 172.16.1.8    10.1.57.2    NULL/1030
14 * > 172.16.1.9    10.1.57.2    NULL/1031
15 * > 172.16.1.10   10.1.57.2    NULL/1032
16 * > 172.16.1.11   10.1.57.2    NULL/1033
17 * > 172.16.1.20   10.1.35.1    1031/NULL
```

ASBR3

```
1 [ASBR3]disp bgp routing-table label
2 Total Number of Routes: 15          分配/学习
3      Network      NextHop      In/Out Label
4
5 * > 172.16.1.1    10.1.57.1    NULL/1030
6 * > 172.16.1.2    10.1.79.2    1034/NULL
7 * i              172.16.1.8
8 * > 172.16.1.3    10.1.57.1    NULL/1029
9 * > 172.16.1.4    10.1.57.1    NULL/1032
10 * > 172.16.1.5    10.1.57.1    NULL/1034
11 * > 172.16.1.6    10.1.57.1    NULL/1033
12 * > 172.16.1.7    127.0.0.1    1035/NULL
13 * i             172.16.1.8
14 * > 172.16.1.8    10.1.78.2    1030/NULL
15 * > 172.16.1.9    10.1.79.2    1031/NULL
16 * > 172.16.1.10   10.1.79.2    1032/NULL
17 * i             172.16.1.8
18 * > 172.16.1.11   10.1.79.2    1033/NULL
19 * > 172.16.1.20   10.1.57.1    NULL/1031
```

B. ASBR 重分配 label, 继续通告给本端 RR (由 RR 反射给 PE 设备)

ASBR1 配置

```
1 route-policy RR permit node 10
2   if-match mpls-label
3   apply mpls-label
4 bgp 100
5   peer 172.16.1.3 route-policy RR export
6   peer 172.16.1.3 label-route-capability
7   # 隐藏 next-hop-local
```

ASBR3 配置

```
1 route-policy RR permit node 10
2   if-match mpls-label
3   apply mpls-label
4 bgp 200
5   peer 172.16.1.9 route-policy RR export
6   peer 172.16.1.9 label-route-capability
```

ASBR2 配置

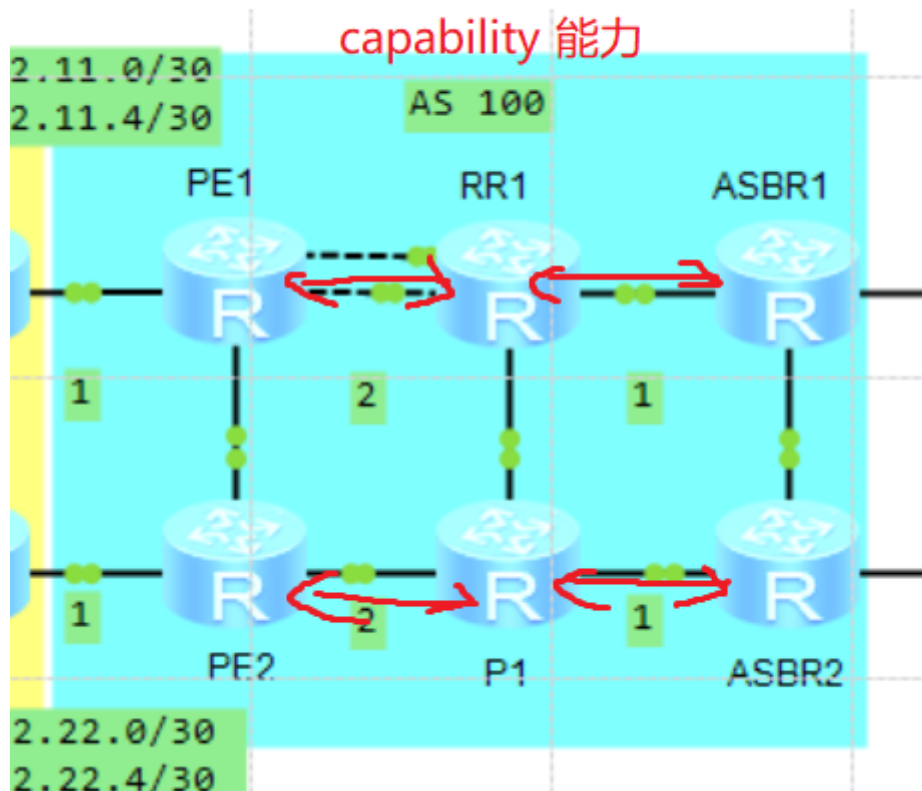
```
1 route-policy RR permit node 10
2   if-match mpls-label
3   apply mpls-label
4 bgp 100
5   peer 172.16.1.3 route-policy RR export
6   peer 172.16.1.3 label-route-capability
7   # 隐藏 next-hop-local
```

ASBR4 配置

```
1 route-policy RR permit node 10
2   if-match mpls-label
3   apply mpls-label
4 bgp 200
5   peer 172.16.1.9 route-policy RR export
6   peer 172.16.1.9 label-route-capability
```

C. RR、ASBR 和 PE 之间都需要开启 标签通告能力

开启 label-route-capability 能力



RR1 配置

```

1  bgp 100
2  peer 172.16.1.5 label-route-capability
3  peer 172.16.1.6 label-route-capability
4  peer 172.16.1.1 label-route-capability
5  peer 172.16.1.20 label-route-capability

```

PE1、PE2 配置

```

1  bgp 100
2  peer 172.16.1.3 label-route-capability

```

RR2 配置

```

1  bgp 200
2  peer 172.16.1.7 label-route-capability
3  peer 172.16.1.8 label-route-capability
4  peer 172.16.1.11 label-route-capability
5  peer 172.16.1.2 label-route-capability

```

PE3、PE4 配置

```

1  bgp 200
2  peer 172.16.1.9 label-route-capability

```

但此时 RR 上查看 bgp-table 还是没有，怎么回事呢？发现 ASBR 之间的接口没有开启 mpls 功能 --- 建议放在 IP-TRUNK 1 那里直接配了



ASBR1-4

```
1 int g0/0/2
2 mpls
```

此时 ASBR、RR 上再查看

以 ASBR1、RR1 为例

已经可以看到 为对端路由 重新分配标签了

```
1 [ASBR1]disp bgp routing-table label
2      Network      NextHop      In/Out Label
3 *> 172.16.1.1      10.1.35.1      1030/NULL
4 * i 172.16.1.6      172.16.1.6
5 *> 172.16.1.2      10.1.57.2      1036/1034
6 *> 172.16.1.3      10.1.35.1      1029/NULL
7 * i 172.16.1.6      172.16.1.6
8 *> 172.16.1.4      10.1.35.1      1032/NULL
9 * i 172.16.1.6      172.16.1.6
10 *> 172.16.1.5      127.0.0.1      1034/NULL
11 * i 172.16.1.6      172.16.1.6
12 *> 172.16.1.6      10.1.56.2      1033/NULL
13 *> 172.16.1.7      10.1.57.2      1035/1035
14 *> 172.16.1.8      10.1.57.2      1040/1030
15 *> 172.16.1.9      10.1.57.2      1039/1031
16 *> 172.16.1.10     10.1.57.2      1038/1032
17 *> 172.16.1.11     10.1.57.2      1037/1033
18 *> 172.16.1.20     10.1.35.1      1031/NULL
19 * i 172.16.1.6      172.16.1.6
```

RR1 也收到了对端的路由了

```
1 [RR1-bgp]disp bgp routing-table label
2      Network      NextHop      In/Out Label
3
4 *>i 172.16.1.2      172.16.1.5      NULL/1036
5 *>i 172.16.1.7      172.16.1.5      NULL/1035
6 *>i 172.16.1.8      172.16.1.5      NULL/1040
7 *>i 172.16.1.9      172.16.1.5      NULL/1039
8 *>i 172.16.1.10     172.16.1.5      NULL/1038
9 *>i 172.16.1.11     172.16.1.5      NULL/1037
```

这里是关键现象

PE1

```
1 [PE1]disp bgp routing-table label
2 Total Number of Routes: 6
3      Network          NextHop          In/Out Label
4
5      i    172.16.1.2    172.16.1.6    NULL/1041
6      i    172.16.1.7    172.16.1.5    NULL/1040
7      i    172.16.1.8    172.16.1.6    NULL/1037
8      i    172.16.1.9    172.16.1.5    NULL/1042
9      i    172.16.1.10   172.16.1.6    NULL/1039
10     i    172.16.1.11   172.16.1.5    NULL/1044
```

PE2

```
1 [PE2]disp bgp routing-table label
2 Total Number of Routes: 6
3      Network          NextHop          In/Out Label
4
5      i    172.16.1.2    172.16.1.6    NULL/1041
6      i    172.16.1.7    172.16.1.5    NULL/1040
7      i    172.16.1.8    172.16.1.6    NULL/1037
8      i    172.16.1.9    172.16.1.5    NULL/1042
9      i    172.16.1.10   172.16.1.6    NULL/1039
10     i    172.16.1.11   172.16.1.5    NULL/1044
```

发现路由为 **i** 不是最优路由。

D. AS100 中因为存在 Level-1 设备没有办法学习到 ASBR1、ASBR2 的明细路由，导致 BGP 标签路由无效，需要配置路由泄露

RR1、P1配置

```
1 ip ip-prefix 172 permit 172.16.0.0 16 greater-equal 32 less-equal 32
2 isis 1
3 import-route isis level-2 into level-1 filter-policy ip-prefix 172
```

验证

此时跨域以及完成了，CE1 --- CE3， CE2 --- CE4， CE1 --- CE4， CE2 --- CE3

```
1 [CE3]ping -a 172.17.1.3 172.17.1.1
2 PING 172.17.1.1: 56 data bytes, press CTRL_C to break
3 Reply from 172.17.1.1: bytes=56 Sequence=1 ttl=250 time=150 ms
4 Reply from 172.17.1.1: bytes=56 Sequence=2 ttl=250 time=120 ms
5 Reply from 172.17.1.1: bytes=56 Sequence=3 ttl=250 time=130 ms
6 Reply from 172.17.1.1: bytes=56 Sequence=4 ttl=250 time=130 ms
7 Reply from 172.17.1.1: bytes=56 Sequence=5 ttl=250 time=100 ms
```

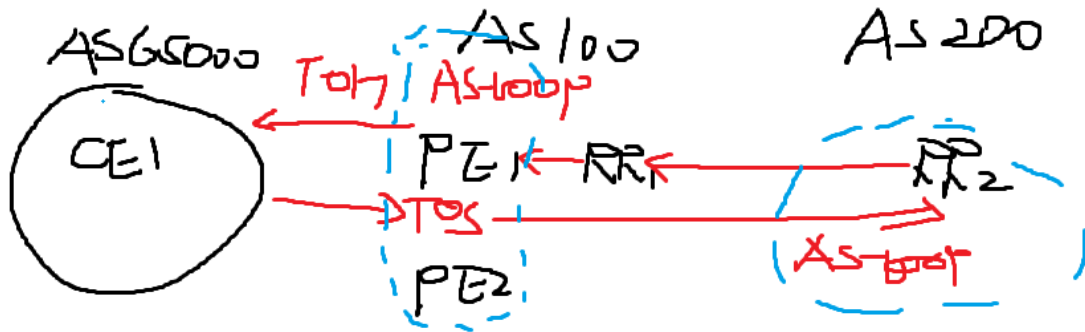
但是我要访问的是 分部 CE3 --- CE4

```

1 [CE3]ping -a 172.17.1.3 172.17.1.4
2   PING 172.17.1.4: 56 data bytes, press CTRL_C to break
3     Request time out
4     Request time out
5     Request time out
6     Request time out
7     Request time out

```

E. 实现 Hub-Spoke 路由正确学习，配置允许 AS 号重复



PE1 配置

```

1 bgp 100
2   ipv4-family vpn-instance TOS
3     peer 10.2.11.6 allow-as-loop

```

PE2 配置

```

1 bgp 100
2   ipv4-family vpn-instance TOS
3     peer 10.2.22.6 allow-as-loop

```

RR2 配置

```

1 bgp 200
2   ipv4-family vpnv4
3     peer 172.16.1.3 allow-as-loop

```

CE3 --- CE4

```

1 ping -a 172.17.1.3 172.17.1.4
2   PING 172.17.1.4: 56 data bytes, press CTRL_C to break
3     Reply from 172.17.1.4: bytes=56 Sequence=1 ttl=250 time=200 ms
4     Reply from 172.17.1.4: bytes=56 Sequence=2 ttl=250 time=210 ms
5     Reply from 172.17.1.4: bytes=56 Sequence=3 ttl=250 time=190 ms
6     Reply from 172.17.1.4: bytes=56 Sequence=4 ttl=250 time=210 ms
7     Reply from 172.17.1.4: bytes=56 Sequence=5 ttl=250 time=230 ms

```

发现通了，tracert 看一下，发现有断层

```

1  tracert -a 172.17.1.3 172.17.1.4
2    1 10.3.33.1 30 ms  1 ms  10 ms
3    2 * * *
4    3 10.1.79.1 100 ms  100 ms  110 ms
5    4 10.1.57.1 130 ms  100 ms  110 ms
6    5 * * *
7    6 10.2.11.5 100 ms  110 ms  110 ms
8    7 10.2.11.6 110 ms  90 ms  100 ms
9    8 10.2.11.1 110 ms  100 ms  110 ms
10   9 10.3.34.2 190 ms  190 ms  190 ms

```

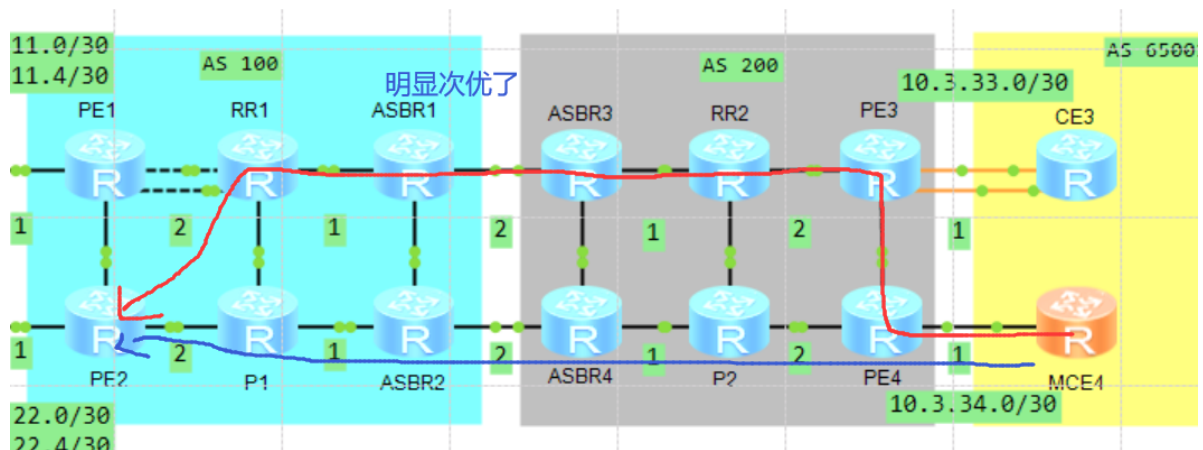
CE4 --- CE2

```

1  tracert -a 172.17.1.4 172.17.1.2
2    1 10.3.34.1 30 ms  20 ms  10 ms
3    2 10.1.102.1 140 ms  100 ms  140 ms
4    3 * * *
5    4 10.1.79.1 110 ms  160 ms  110 ms
6    5 10.1.57.1 120 ms  130 ms  130 ms
7    6 * * *
8    7 10.1.13.1 120 ms  140 ms  140 ms
9    8 10.2.22.5 110 ms  130 ms  130 ms
10   9 10.2.22.6 140 ms  110 ms  130 ms

```

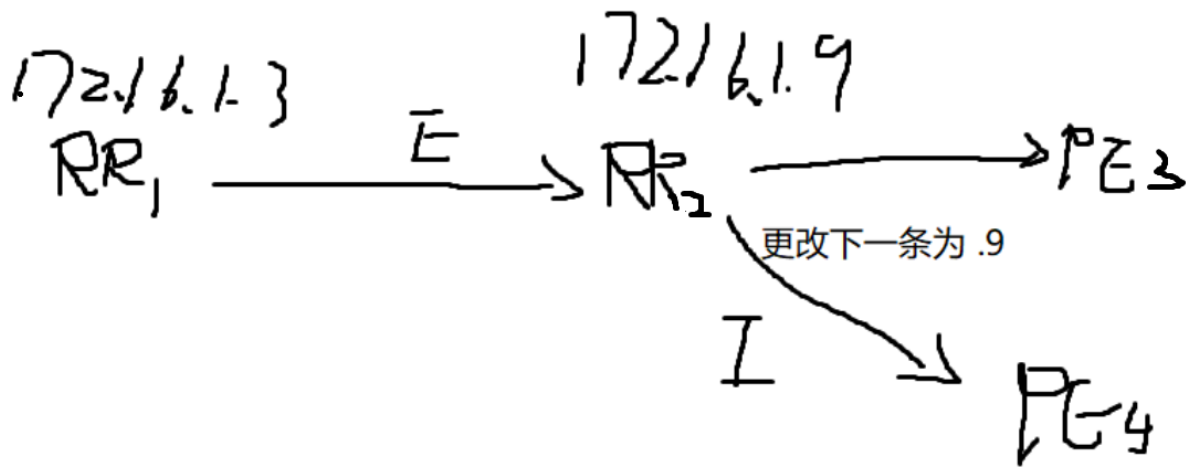
发现次优了



发现中间断了，这是没有建立完整的 LSP 通道的原因

F. MPLS 域内不能出现次优问题！

MPLS-VPN 传递路由 EBGp 传递给 IBGP 时候会自动更改下一跳，所以额外需要配置下一跳不变



RR1 配置

```

1 bgp 100
2  ipv4-family vpnv4
3     peer 172.16.1.9 next-hop-invariable
4     peer 172.16.1.1 next-hop-invariable
5     peer 172.16.1.20 next-hop-invariable

```

RR2 配置

```

1 bgp 200
2  ipv4-family vpnv4
3     peer 172.16.1.3 next-hop-invariable
4     peer 172.16.1.11 next-hop-invariable
5     peer 172.16.1.2 next-hop-invariable

```

在验证前建议在 PE 上配置：

用来使能 MPLS VPN 报文的 IP TTL复制功能

```

1 mpls
2  ttl propagate vpn

```

验证：完美的 15 跳

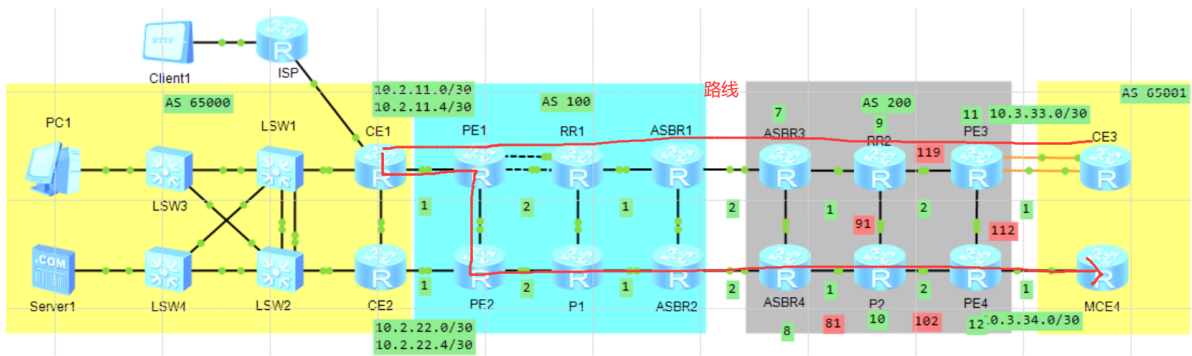
```

1 [PE3]tracert -a 172.17.1.3 172.17.1.4
2 traceroute to 172.17.1.4(172.17.1.4), max hops: 30 ,packet length: 40,press
CT
3 RL_C to break
4  1 10.3.33.1 20 ms  20 ms  20 ms
5  2 10.1.119.1 110 ms  100 ms  110 ms
6  3 10.1.79.1 100 ms  110 ms  110 ms
7  4 10.1.57.1 120 ms  130 ms  100 ms
8  5 10.1.35.1 70 ms  100 ms  110 ms
9  6 10.2.11.5 110 ms  100 ms  110 ms
10 7 10.2.11.6 110 ms  100 ms  110 ms # CE1
11 8 10.2.11.1 90 ms  120 ms  80 ms # PE1
12 9 10.1.12.2 210 ms  210 ms  210 ms # PE2
13 10 10.1.24.2 230 ms  210 ms  210 ms # P1
14 12 10.1.68.2 200 ms  250 ms  200 ms
15 13 10.1.81.2 190 ms  210 ms  230 ms
16 14 10.3.34.1 190 ms  210 ms  210 ms

```

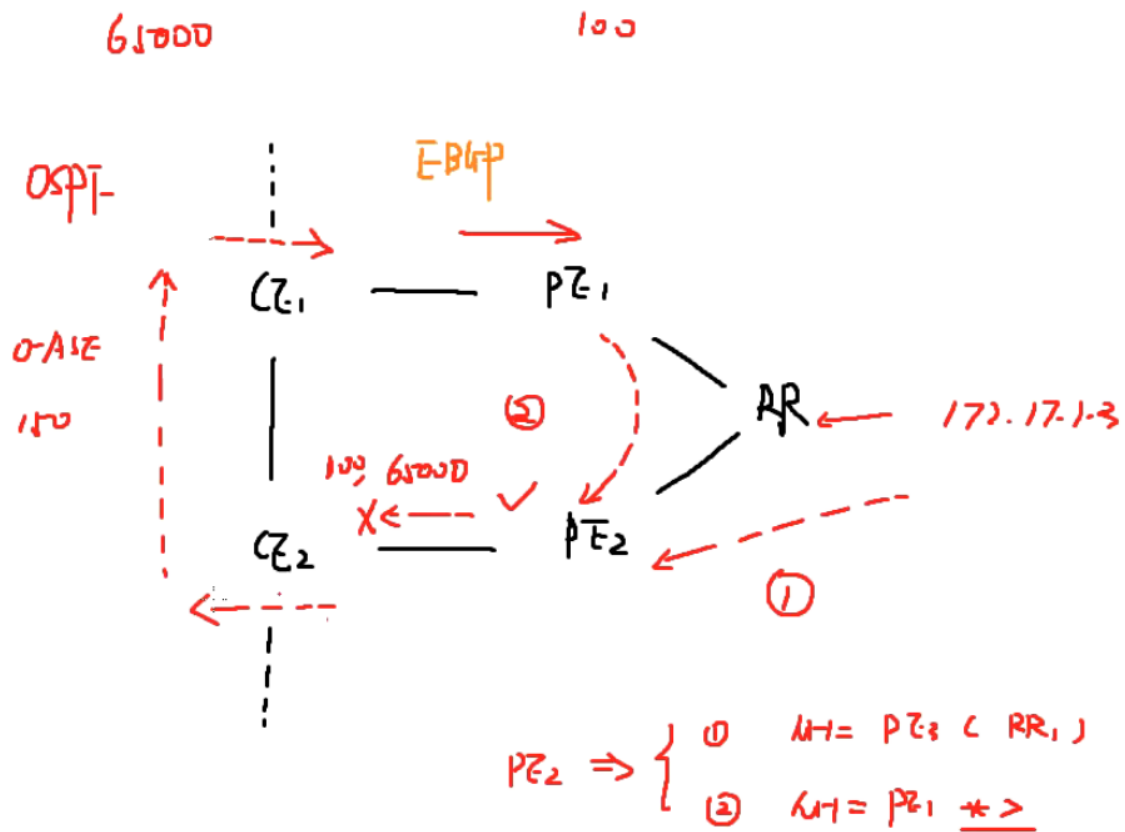
17 15 10.3.34.2 170 ms 240 ms 190 ms

```
1 [CE4]tracert -vpn-instance VPN1 -a 172.17.1.4 172.17.1.3
2 traceroute to VPN1 17
3 2.17.1.3(172.17.1.3), max hops: 30 ,packet length: 40,press CTRL_C to break
4 1 10.3.34.1 20 ms 20 ms 20 ms
5 2 10.1.102.1 120 ms 130 ms 120 ms
6 3 10.1.81.1 160 ms 100 ms 130 ms
7 4 10.1.68.1 120 ms 130 ms 150 ms
8 5 10.1.46.1 130 ms 130 ms 110 ms
9 6 10.2.22.5 120 ms 110 ms 120 ms
10 7 10.2.22.6 120 ms 130 ms 130 ms # CE2
11 8 10.2.22.1 120 ms 130 ms 120 ms # PE2
12 9 10.1.12.1 230 ms 230 ms 230 ms # PE1
13 10 10.1.13.2 200 ms 240 ms 250 ms # RR1
14 11 10.1.35.2 200 ms 250 ms 220 ms
15 12 10.1.57.2 260 ms 220 ms 270 ms
16 13 10.1.79.2 220 ms 200 ms 220 ms
17 14 10.3.33.1 210 ms 200 ms 260 ms
18 15 10.3.33.2 230 ms 260 ms 210 ms
```



9. CE1-PE1之间链路开，CE1设备仍可以学习到spoke业务网段。配置保障有最好的扩展性。（6）
10. 在拓扑正常情况下，要求CE1，CE2访问spoke网段时，不从本AS内绕行。（1）

分析



CE 上 bgp 和 ospf 双向引入，不会导致环路问题，但是会导致路由回灌-路由震荡问题，次优问题

不会导致环路问题：从 RR2 传来了 172.17.1.3/32 路由，通过 RR1 反射给 PE2 - CE2 - CE1 - PE1 - RR1 - PE2 - CE2，到 CE2 的时候 AS 号为 100 65000，因为有 65000，所以 CE2 不收，所以不会导致环路问题

路由震荡问题：

① 号路由为正常从 RR2 传来了 172.17.1.3/32 路由，通过 RR1 反射给 PE2 nh = PE3 (RR1 传来)

② 号路由为从 RR2 传来了 172.17.1.3/32 路由，通过 RR1 反射给 PE2 - CE2 - CE1 - PE1 - RR1 - PE2

nh = PE1

PE2 视角

序号	路由	下一跳	路径	as-path	优选
1	172.17.1.3/32	172.16.1.3(PE3)	RR1 - PE2	100	
2	172.17.1.3/32	172.16.1.1(PE1)	PE2 - CE2 - CE1 - PE1 - RR1 - PE2	65000, 100	Y (离得更近)

此时 PE2 会选路，as 号都是一样的，比较 nh 哪个离自己更近 (igp 开销小)，此时会认为是 PE1 的 ②号路由更近

然后 ②号路由传给 CE2，CE2 不收，此时 ②号路由不传了，①号路由占上风了，然后就这么来来回回翻动

解决方案：给路由打 tag

注意：需要在 PE1 - RR1 的 ip-trunk 1 链路加上 isis cost 配置，否则后续 CE4 - CE3 选路不正确

```
1 int Ip-Trunk1
2   isis cost 1500
```

CE1 配置

```
1 ospf
2   import-route bgp tag 172
3
4 route-policy O2B deny node 10
5   if-match tag 172
6 route-policy O2B permit node 20
7
8 bgp 65000
9   import-route ospf 1 route-policy O2B
```

CE2 配置

```
1 route-policy O2B deny node 10
2   if-match tag 172
3 route-policy O2B permit node 20
4
5 bgp 65000
6   import-route ospf 1 route-policy O2B
7
8 ospf
9   import-route bgp tag 172
```

修改 EBGp 优先级

CE1、CE2

```
1 bgp 65000
2   preference 120 255 255
3   # EBGp IBGP LOCAL
```

此处为什么将 EBGp 优先级设置为 120，是因为需要比 O_ASE 优先级 150 低就行了。

```
1 [CE1]disp ip routing-table 172.17.1.3
2 -----
3
4 Destination/Mask    Proto   Pre  Cost   Flags NextHop         Interface
5 172.17.1.3/32      O_ASE   150   1       D    10.2.12.2
6 GigabitEthernet
7 0/0/0
```

验证：

```

1 [CE1-bgp]disp ip routing-table 172.17.1.3
2 -----
3 Destination/Mask   Proto   Pre  Cost   Flags NextHop         Interface
4      172.17.1.3/32 EBGP    120   0       D    10.2.11.1
5      GigabitEthernet
6      0/0/1.1

```

验证：CE1、CE2 下查看到有 deny

```

1 [CE1]disp route-policy
2 Route-policy : 02B
3   deny : 10 (matched counts: 5)
4   Match clauses :
5     if-match tag 172
6   permit : 20 (matched counts: 5)

```

7. 奇偶路由选路

11. 在PE3, PE4上修改BGP local-preference属性为120, 实现CE3,CE4访问非直接的10.3.x.0/24网段时,

若X为奇数, PE3, PE4优选的下一跳为 PE1;

若X为偶数, PE3, PE4优选的下一跳为PE2, 不用考虑来回路径是否一致。(3分)

PE3、PE4 配置

```

1 10.3.1.x PE1
2 10.3.2.x PE2

```

配置思路

通过修改 local-preference 来选路

如果是 奇数路由, 且下一跳为 PE1(172.16.1.1)则修改 local-preference 为 120

如果是 偶数路由, 且下一跳为 PE2(172.16.1.20)则修改 local-preference 为 120

1. 先用 acl 匹配出 奇偶路由

```

1 acl 2001
2   rule 5 permit source 10.3.1.0 0.0.254.0
3 acl 2002
4   rule 5 permit source 10.3.0.0 0.0.254.0

```

2. 再用 ip-prefix 匹配出 PE1、PE2

```

1 ip ip-prefix PE1 permit 172.16.1.1 32
2 ip ip-prefix PE2 permit 172.16.1.20 32

```

3. 通过 route-policy 修改 local-preference

```

1 route-policy LP permit node 10
2   if-match acl 2001
3   if-match ip next-hop ip-prefix PE1
4   apply local-preference 120
5 route-policy LP permit node 20
6   if-match acl 2002
7   if-match ip next-hop ip-prefix PE2
8   apply local-preference 120
9 route-policy LP permit node 30

```

4. 对 RR2 入向调用

```

1 bgp 200
2   ipv4-family vpnv4
3     peer 172.16.1.9 route-policy LP import

```

供整体复制：

```

1 ip ip-prefix PE1 permit 172.16.1.1 32
2 ip ip-prefix PE2 permit 172.16.1.20 32
3 acl 2001
4   rule 5 permit source 10.3.1.0 0.0.254.0
5 acl 2002
6   rule 5 permit source 10.3.0.0 0.0.254.0
7 route-policy LP permit node 10
8   if-match acl 2001
9   if-match ip next-hop ip-prefix PE1
10  apply local-preference 120
11 route-policy LP permit node 20
12   if-match acl 2002
13   if-match ip next-hop ip-prefix PE2
14  apply local-preference 120
15 route-policy LP permit node 30
16 bgp 200
17   ipv4-family vpnv4
18     peer 172.16.1.9 route-policy LP import

```

验证：

PE3、PE4 上查看

```

1 disp bgp vpnv4 all routing-table
2  *>i 10.3.1.0/24      172.16.1.1      120      0      100
3  65000?
4  * i                172.16.1.20      100      0      100
5  65000?
6  *>i 10.3.2.0/24      172.16.1.20      120      0      100
7  65000?
8  * i                172.16.1.1      100      0      100
9  65000?

```

4.Future (17分)

4.1 HA (8分)

1. CE1配置静态的默认路由访问ISP，下一跳IP为100.0.1.2，该默认路由要与CE1-ISP链路的BFD状态绑定（CE1的对端设备不支持BFD），感受故障时间要小于150ms（2）

1. 配置 CE1 单臂回声

```
1 bfd
2 q
3 bfd toisp bind peer-ip 100.0.1.2 interface GigabitEthernet2/0/0 one-arm-echo
4 discriminator local 1 # 配置本地标识符
5 commit
```

验证：已 up

```
1 [CE1]disp bfd session all
2 -----
3 Local Remote PeerIpAddr State Type InterfaceName
4 -----
5
6 1 - 100.0.1.2 Up S_IP_IF GigabitEthernet2/0/0
7 -----
8 Total UP/DOWN session Number : 1/0
```

课后问题：怎么构造 upd 报文？抓包观察，源目 ip地址，端口号

2. 绑定 BFD 会话

CE1

```
1 ip route-static 0.0.0.0 0 100.0.1.2 track bfd-session toisp
```

验证：

```
1 [CE1]disp ip rou pro static
2 Route Flags: R - relay, D - download to fib
3 -----
4 Static routing table status : <Active>
5 Destinations : 1 Routes : 1
6
7 Destination/Mask Proto Pre Cost Flags NextHop Interface
8 0.0.0.0/0 static 60 0 RD 100.0.1.2 GigabitEthernet2/0/0
```

3. 配置感受故障时间 --- 忘记写

CE1

```

1 bfd toisp
2   # 因为故障次数为 3 次, 3 * 40 = 120 ms
3   min-echo-rx-interval 40
4   commit

```

2. CE2, CE3, CE4能够通过默认路由访问ISP (4)

将默认路由通告给 CE2、CE3、CE4

给 CE2 下发默认路由

CE1 配置

```

1 ospf
2   default-route-advertise

```

CE3、CE4 借助 BGP TOS 邻居 通告给 PE1, PE1 借助 MP-BGP 传给 PE3、PE4 默认路由

CE1 配置

```

1 bgp 65000
2   peer 10.2.11.5 default-route-advertise

```

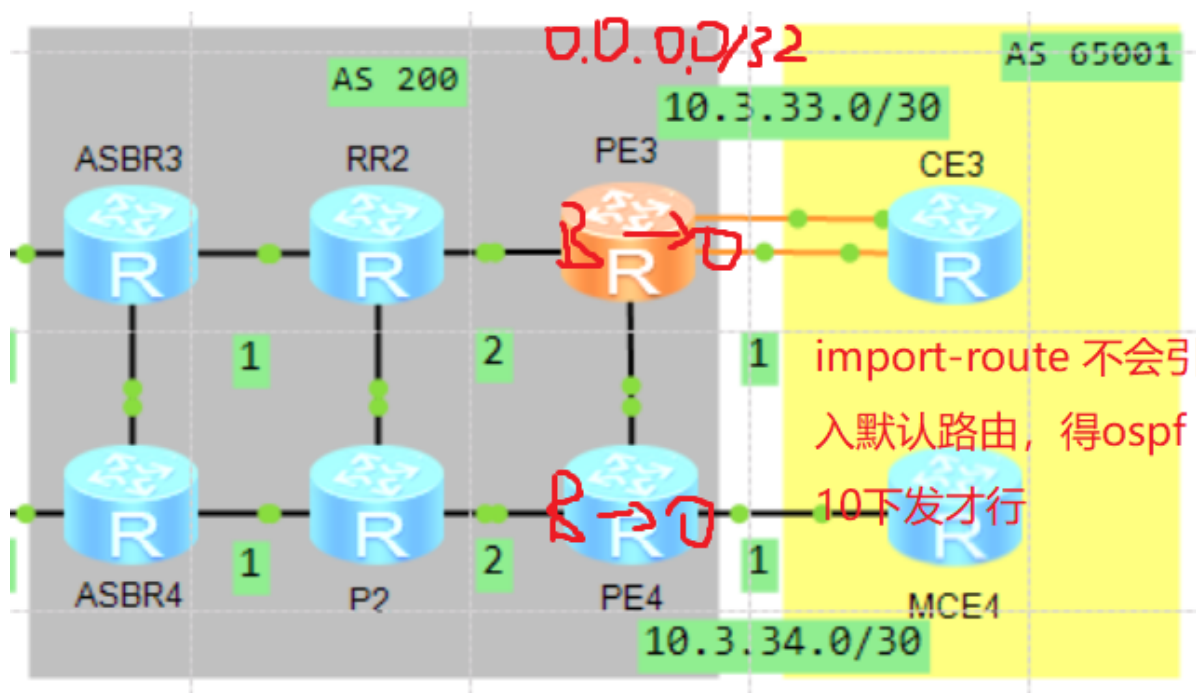
PE3、PE4

```

1 <PE4>disp ip rou vpn-instance VPN1 0.0.0.0
2 Route Flags: R - relay, D - download to fib
3 -----
4 Routing Table : VPN1
5 Summary Count : 1
6 Destination/Mask   Proto  Pre  Cost    Flags NextHop         Interface
7      0.0.0.0/0      IBGP   255  0       RD    172.16.1.1
8      GigabitEthernet
9      0/0/2

```

但是此时 CE3、CE4 上是没有默认路由的, 原因是因为



PE3

```
1 ospf 10
2 default-route-advertise
```

PE4

```
1 ospf 20
2 default-route-advertise
```

查看

CE3、CE4

```
1 <CE3>disp ip rou 0.0.0.0
2 Route Flags: R - relay, D - download to fib
3 -----
4 --
5 Destination/Mask    Proto  Pre  Cost    Flags NextHop        Interface
6      0.0.0.0/0      O_ASE  150  1        D   10.3.33.1          Mp-
7 group0/0/0
8
9 <CE4>disp ip rou 0.0.0.0
10 Route Flags: R - relay, D - download to fib
11 -----
12 --
13 Destination/Mask    Proto  Pre  Cost    Flags NextHop        Interface
14      0.0.0.0/0      O_ASE  150  1        D   10.3.34.1          GigabitEthernet0/0/1
15
16 <MCE4>disp ip rou vpn-instance VPN1 0.0.0.0
17 Route Flags: R - relay, D - download to fib
18 -----
19 --
20 Destination/Mask    Proto  Pre  Cost    Flags NextHop        Interface
```

```

17      0.0.0.0/0    O_ASE    150    1          D    10.3.34.1
    GigabitEthernet
18    0/0/1

```

补充:

CE2 可以配置 BGP 传给 PE2

```

1  bgp 65000
2  peer 10.2.22.5 default-route-advertise

```

4.2 NAT (2分)

1. 在CE1上, 10.3.0.0/16 (不含10.3.2.10) 的内网地址转换为102.0.1.2-102.0.1.6, 通过Ge2/0/0访问ISP。server1拥有单独的公网地址 102.0.1.1, 对ISP提供FTP和HTTP (2)

解法: 注意: 此处NAT映射的公网地址通常称为业务地址 (102.0.1.2-6), 与CE1和ISP接口互联地址 (100.0.1.0/30) 不一致

CE1

```

1  undo int lo1
2  acl 2000
3      rule 5 deny source 10.3.2.10 0
4      rule 10 permit source 10.3.0.0 0.0.255.255
5      q
6  nat address-group 1 102.0.1.2 102.0.1.6
7  interface GigabitEthernet2/0/0
8      ip address 100.0.1.1 255.255.255.252
9      nat outbound 2000 address-group 1
10 interface GigabitEthernet2/0/0
11     nat server protocol tcp global 102.0.1.1 ftp inside 10.3.2.10 ftp
12     nat server protocol tcp global 102.0.1.1 www inside 10.3.2.10 www

```

验证: 在家练习的时候可以在ISP添加回包静态路由, 访问ISP的8.8.8.8

ISP

```

1  ip route-static 102.0.1.0 29 100.0.1.1

```

考试ISP不能登录 (虽然密码为hcie 一登就登上了), 但是不用测试, 保证NAT配置无误即可

pc1

```

1  PC>ping 8.8.8.8

```

No.	Time	Source	Destination	Protocol	Length	Info
461	13.735000	102.0.1.6	8.8.8.8	ICMP	74	Echo (ping) request id=0x0528, seq=
462	13.735000	8.8.8.8	102.0.1.6	ICMP	74	Echo (ping) reply id=0x0528, seq=
499	14.797000	102.0.1.6	8.8.8.8	ICMP	74	Echo (ping) request id=0x0628, seq=
500	14.797000	8.8.8.8	102.0.1.6	ICMP	74	Echo (ping) reply id=0x0628, seq=
537	15.860000	102.0.1.6	8.8.8.8	ICMP	74	Echo (ping) request id=0x0728, seq=

CE3

```

1 [CE3]ping -a 172.17.1.3 8.8.8.8
2   PING 8.8.8.8: 56 data bytes, press CTRL_C to break
3     Reply from 8.8.8.8: bytes=56 Sequence=1 ttl=248 time=110 ms
4     Reply from 8.8.8.8: bytes=56 Sequence=2 ttl=248 time=100 ms
5     Reply from 8.8.8.8: bytes=56 Sequence=3 ttl=248 time=110 ms
6     Reply from 8.8.8.8: bytes=56 Sequence=4 ttl=248 time=110 ms
7     Reply from 8.8.8.8: bytes=56 Sequence=5 ttl=248 time=130 ms

```

问题：为什么 CE1 上没有接口是 102.0.1.0 网段的，那为啥可以，

因为在配置 `nat outbound 2000 address-group 1` 的时候，会自动生成用户自定义的路由条目

```

1 [CE1]disp ip routing | include Unr
2   102.0.1.1/32 Unr    64  0          D   127.0.0.1      InLoopBack0
3   102.0.1.2/32 Unr    64  0          D   127.0.0.1      InLoopBack0
4   102.0.1.3/32 Unr    64  0          D   127.0.0.1      InLoopBack0
5   102.0.1.4/32 Unr    64  0          D   127.0.0.1      InLoopBack0
6   102.0.1.5/32 Unr    64  0          D   127.0.0.1      InLoopBack0
7   102.0.1.6/32 Unr    64  0          D   127.0.0.1

```

4.3 QOS (7分)

1. 在CE1和G2/0/0的出方向，周一至周五的8:00-18:00点对TCP目的端口号6881-6999流量，承诺平均速率为1Mbps (3)

流量监管

CE1

定义一个时间

```

1 time-range w 08:00 to 18:00 working-day

```

在用 高级 acl 匹配流量

```

1 acl 3000
2 rule permit tcp destination-port range 6881 6999 time-range w

```

CE1 的出方向 g2/0/0 进行限速

```

1 int g2/0/0
2 qos car outbound acl 3000 cir 1000

```

这一步不需要验证

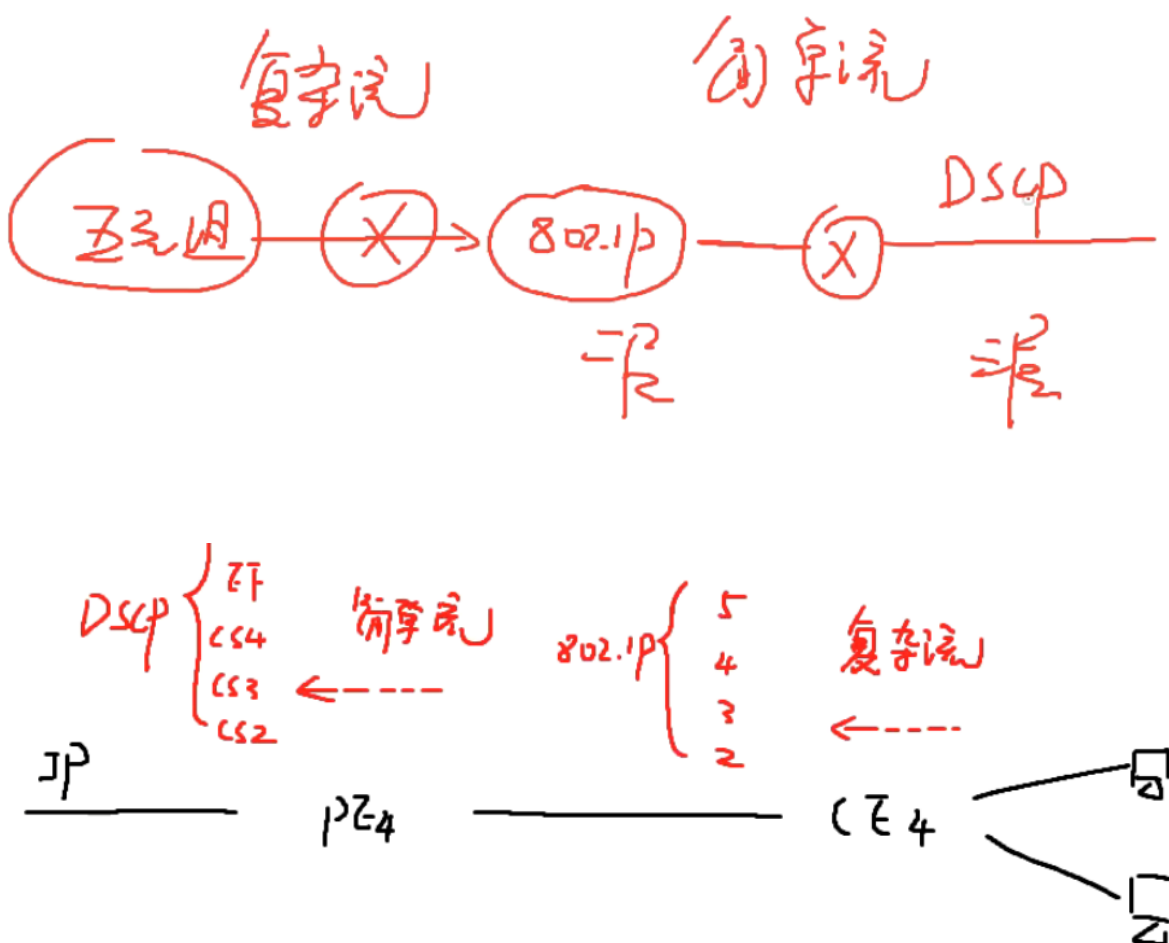
2. CE4-PE4的QOS规划如下表所示：

业务地址前缀（目标）	业务类别	802.1p	DSCP	队列调度		拥塞避免			
				调度策略	Weight值	拥塞传递机制	低向限	高向限	丢包概率
10.3.1.0/24	Real time	101	EF	PQ		不			
10.3.2.0/24	Signal	100	CS4	WFQ	63	WRED	70%	100%	50%
10.3.3.0/24	Monitor	011	CS3	WFQ	21	WRED	50%	90%	50%
10.3.4.0/24	Office	010	CS2	WFQ	9	WRED	50%	80%	50%
其他	BE	000	BE	WFQ	1	WRED	50%	80%	50%

在CE4的G0/0/1出方向对流量进行802.1p标记。在PE4的G0/0/1入方向，继承CE4的802.1p值，并将802.1p映射为Dscp（2）

分类标记

先复杂流（五元组）--- 简单流



1. CE4 配置 --- 复杂流分类（MQC）

1. acl 区别业务网段

```

1  acl 3001
2  rule permit ip destination 10.3.1.0 0.0.0.255
3  acl 3002
4  rule permit ip destination 10.3.2.0 0.0.0.255
5  acl 3003
6  rule permit ip destination 10.3.3.0 0.0.0.255
7  acl 3004
8  rule permit ip destination 10.3.4.0 0.0.0.255

```

2. 创建复杂流分类匹配出流量

```
1 traffic classifier R
2   if-match acl 3001
3 traffic classifier S
4   if-match acl 3002
5 traffic classifier M
6   if-match acl 3003
7 traffic classifier O
8   if-match acl 3004
```

3. 创建流行为标记 802.1p 的值

```
1 traffic behavior R
2   remark 8021p 5
3 traffic behavior S
4   remark 8021p 4
5 traffic behavior M
6   remark 8021p 3
7 traffic behavior O
8   remark 8021p 2
```

4. 创建流策略进行关联流分类和流行为关系

```
1 traffic policy RM
2   classifier R behavior R
3   classifier S behavior S
4   classifier M behavior M
5   classifier O behavior O
```

5. 在 CE4 的 g0/0/1 出方向进行调用

```
1 int g0/0/1
2   traffic-policy RM outbound
```

注意：如果想要看到实验现象的话得需要子接口

2. PE4 配置简单流分类

1. 将 802.1p 的值映射成 dscp

```
1 qos map-table dot1p-dscp
2   input 5 output 46
```

DSCP	PRE					
INTEGER<0-63> DSCP (DiffServ CodePoint) value						
af11	AF11 DSCP (001010)					
af12	AF12 DSCP (001100)					
af13	AF13 DSCP (001110)					
af21	AF21 DSCP (010010)					
af22	AF22 DSCP (010100)					
af23	AF23 DSCP (010110)					
af31	AF31 DSCP (011010)					
af32	AF32 DSCP (011100)					
af33	AF33 DSCP (011110)					
af41	AF41 DSCP (100010)					
af42	AF42 DSCP (100100)					
af43	AF43 DSCP (100110)					
cs1	CS1 (IP Precedence 1) DSCP (001000)	8				
cs2	CS2 (IP Precedence 2) DSCP (010000)	16				
cs3	CS3 (IP Precedence 3) DSCP (011000)	24				
cs4	CS4 (IP Precedence 4) DSCP (100000)	32				
cs5	CS5 (IP Precedence 5) DSCP (101000)					
cs6	CS6 (IP Precedence 6) DSCP (110000)					
cs7	CS7 (IP Precedence 7) DSCP (111000)					
default	Default DSCP (000000)					
ef	EF DSCP (101110)	46				

2. PE4 的入接口 `g0/0/1` 得信任 CE4 传来的 `802.1p`

```

1  int g0/0/1
2  trust 8021p override

```

拥塞管理和拥塞控制

队列技术、丢弃技术

DSCP

Queue

EF → 5

CS 4 → 4

CS 3 → 3

2) CS 2 → 2

Schedule

调度

路由器会自动把 dscp 值 加入到不同的 queue 中，此时需要通过 schedule 技术来管理这些队列

设备上，每个接口出方向都拥有4个或8个队列，以队列索引号进行标识，队列索引号分别为0、1、2、3或0、1、2、3、4、5、6、7。设备根据本地优先级和队列之间的映射关系，自动将分类后的报文流送入各队列，然后按照各种队列调度机制进行调度。下面以每个接口8个队列对各种调度方式进行说明。

1. PQ

这种方式它严格按照队列优先级来，横向一个一个走，这样的缺点就是优先级低的队列容易被饿死

PQ调度，针对于关键业务类型应用设计，PQ调度算法维护一个优先级递减的队列系列并且只有当更高优先级的所有队列为空时才服务低优先级的队列。这样，将关键业务的分组放入较高优先级的队列，将非关键业务（如E-Mail）的分组放入较低优先级的队列，可以保证关键业务的分组被优先传送，非关键业务的分组在处理关键业务数据的空闲间隙被传送。



1. WFQ

这种方式按照 `weight` 方式来，把流量分段，竖向的一个一个走

加权循环调度WRR (Weight Round Robin) 在循环调度RR (Round Robin) 的基础上演变而来，在队列之间进行轮流调度，根据每个队列的权重来调度各队列中的报文流。



`TAIL` 尾丢弃 --- 大家一起死

`RED` 早期随机丢弃 ---

`WRED`

限速技术

流量监管、流量整形 --- 令牌桶算法 评估

3. PE4的G0/0/0和G0/0/2匹配DSCP值，根据表1，配置拥塞管理和拥塞避免。（2）

PE4

1. 创建 `WRED` 丢弃模板 `data`

```
1 drop-profile data
2 # 配置当前WRED丢弃模板基于DSCP优先级进行丢弃
3 wred dscp
4 dscp cs4 low-limit 70 high-limit 100 discard-percentage 50
5 dscp cs3 low-limit 50 high-limit 90 discard-percentage 50
6 dscp cs2 low-limit 50 high-limit 80 discard-percentage 50
7 dscp default low-limit 50 high-limit 80 discard-percentage 50
```

2. 创建队列模板 `queue-profile1`

```

1 qos queue-profile queue-profile1
2   # 配置调度策略
3   schedule pq 5
4   schedule wfq 0 to 4
5   # 配置 权重
6   queue 4 weight 63
7   queue 3 weight 21
8   queue 2 weight 9
9   queue 0 weight 1
10  # 配置 wfq 丢弃模板
11  queue 0 drop-profile data
12  queue 2 to 4 drop-profile data

```

3. 在出接口下应用队列模板 queue-profile1

```

1 int g0/0/0
2   qos queue-profile queue-profile1
3 int g0/0/2
4   qos queue-profile queue-profile1

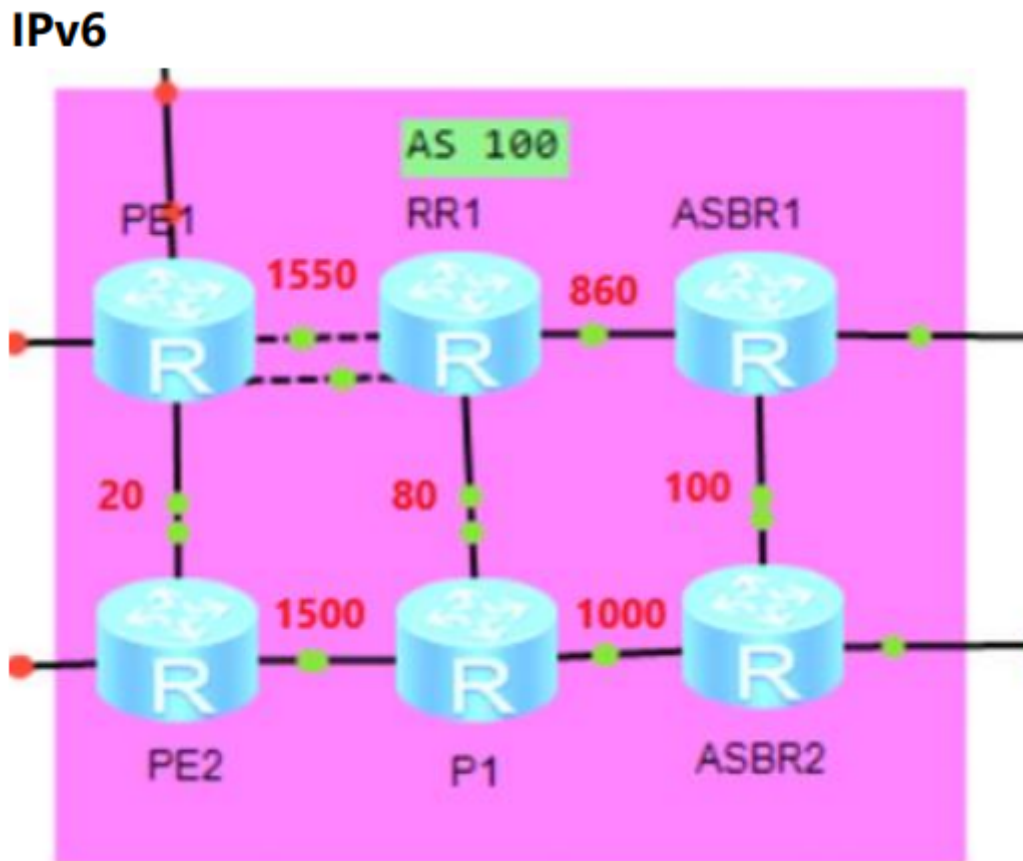
```

5. IPV6 (14分)

5.1 基础配置

5.2 IPV6 ISIS (3)

- 如图6, PE1,PE2,RR1,P1,ASBR1,ASBR2,运行ISIS协议, 各直接网段通告入ISIS, 配置各链路cost (3)



注意: 以上 6 台设备一定要打开 isis 的多拓扑能力

```

1 sy
2 isis
3     ipv6 enable topology ipv6
4     quit

```

以 PE1 为例配置

```

1 interface Ip-Trunk1
2     isis ipv6 enable
3     isis ipv6 cost 1550
4 int g0/0/0
5     isis ipv6 enable
6     isis ipv6 cost 20
7 int LoopBack0
8     isis ipv6 enable

```

检查:

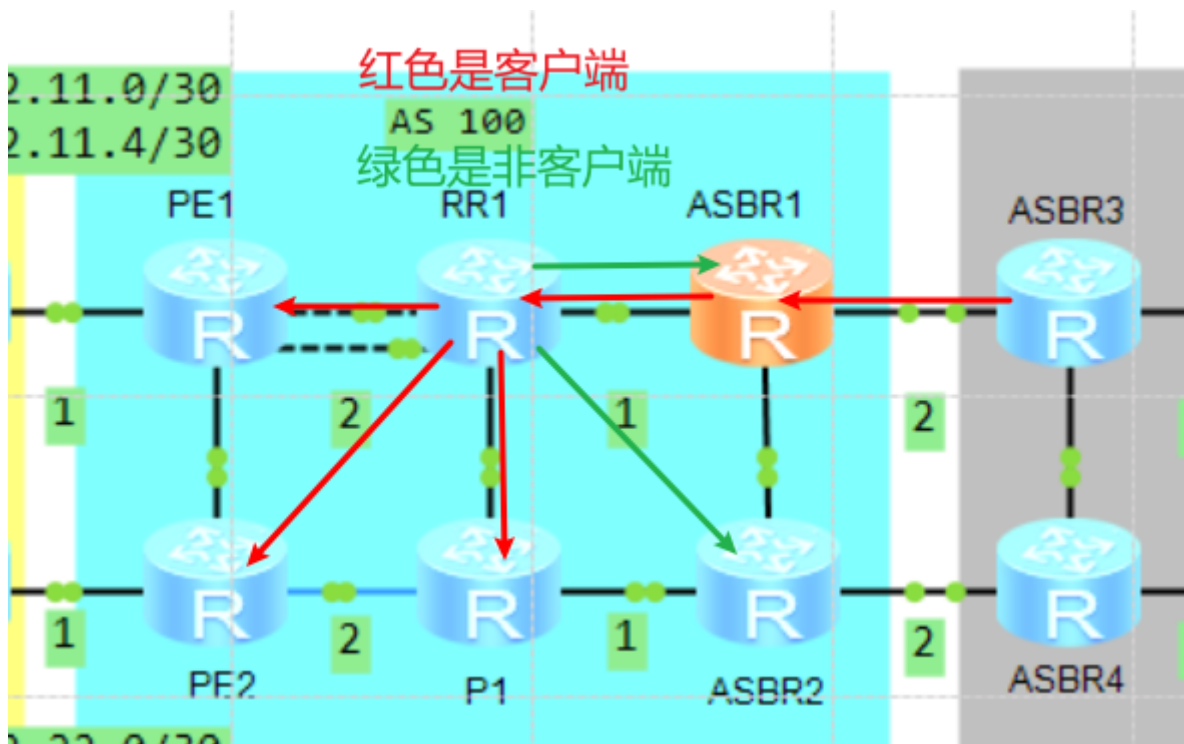
```

1 disp isis int # 看接口 ipv6 up
2 disp ipv6 routing pro isis # 看到 D1 to D6 可以验证下 cost 是否为 2520 2510

```

5.3 IPV6 BGP (11分)

- 如图7, ASBR1-ASBR3通过直接链路建立EBGP4+邻居。PE1,PE2,P1是RR1的IBGP+客户站。(已配)



1. ASBR1、ASBR3 建立 EBGP4+ 需求

ASBR1 配置

建议先在 ASBR1、3 之间查看 `ipv6` 地址, 用于 BGP 建立邻居时候。

```

1  bgp 100
2    peer 2570:CCDD:CCBB:3CAF:EFFE:ACDD:CCDB:5700 as 200
3    ipv6-family unicast
4      peer 2570:CCDD:CCBB:3CAF:EFFE:ACDD:CCDB:5700 enable

```

ASBR3 配置

```

1  bgp 200
2    peer 2570:CCDD:CCBB:3CAF:EFFE:ACDD:CCDB:5701 as 100
3    ipv6-family unicast
4      peer 2570:CCDD:CCBB:3CAF:EFFE:ACDD:CCDB:5701 enable

```

2. PE1、PE2、P1 是 RR1 的 IBGP+

RR1 配置

```

1  bgp 100
2    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC01 as 100
3    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC01 con lo0
4    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC02 as 100
5    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC02 con lo0
6    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC04 as 100
7    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC04 con lo0
8    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 as 100
9    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 con lo0
10   peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC06 as 100
11   peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC06 con lo0
12
13   ipv6-family unicast
14     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC01 enable
15     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC01 reflect-client
16     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC02 enable
17     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC02 reflect-client
18     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC04 enable
19     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC04 reflect-client
20     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 enable
21     peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC06 enable

```

PE1、PE2、P1、ASBR1、ASBR2 配置

```

1  bgp 100
2    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC03 as 100
3    peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC03 con lo0
4    ipv6-family unicast
5      peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC03 enable

```

验证:


```

1 # ASBR3 检查 EBGp4+
2 disp bgp ipv6 peer
3 # RR 检查 IBGP+
4 # RR1 上检查
5 disp bgp ipv6 peer
6 # 有 5 个邻居
7 # 发现有 DC01 - DC06 就行

```

若此时邻居关系无法建立，可能因为各个设备 LoopBack0 未开启 `isis ipv6 enable`

2. 在ASBR1将ISIS IPV6的路由导入BGP4+,只向ASBR3通告前缀为 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00/120的路由（不能使用route-policy）。将ASBR3的loopback0通告入BGP4+ (4)

A、在ASBR1将ISIS IPV6的路由导入BGP4+

ASBR1 配置

```

1 bgp 100
2   ipv6-family unicast
3     import-route isis 1

```

ASBR3 查看 --- 能看到 13 条就完事了

```

1 [ASBR3]disp bgp ipv6 routing-table
2
3 BGP Local router ID is 10.1.78.1
4 Status codes: * - valid, > - best, d - damped,
5               h - history, i - internal, s - suppressed, S - stale
6               Origin : i - IGP, e - EGP, ? - incomplete
7
8 Total Number of Routes: 13

```

B、只向ASBR3通告前缀为2000\EAD8:99EF\C03E:B2AD:9EFF:32DD:DC00/120的路由（不能使用route-policy）

注意：ipv6 没有自动汇总，需手工配置 `aggregate`

ASBR1 配置

```

1 bgp 100
2   ipv6-family unicast
3     aggregate 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 120 detail-suppressed #
      抑制明细

```

此时 ASBR3 查看 就只有一条汇总路由了

```

1 [ASBR3]disp bgp ipv6 routing-table
2
3 BGP Local router ID is 10.1.78.1
4 Status codes: * - valid, > - best, d - damped,
5                h - history, i - internal, s - suppressed, S - Stale
6                Origin : i - IGP, e - EGP, ? - incomplete
7
8 Total Number of Routes: 1
9 *> Network   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
10    NextHop   : 2570:CCDD:CCBB:3CAF:EFBE:ACDD:CCDB:5701 LocPrf      :
11
12    MED       :                                         PrefVal   : 0
13
14    Label     :
15    Path/Ogn  : 100 ?

```

是不是此时的现象很爽？是因为 PE RR ASBR 之间的 ipv6 互通地址前缀，正好在你的 /120 位前缀内，所以一并被汇总了

注意：考场直连网段可能不在我们的汇总的路由范围内，所以需要通过 路由过滤 实现只发送汇总路由

此时模拟 ipv6 互通地址并不能正好被 /120 合并

PE1 和 PE2 的 g0/0/0 的 ipv6 地址改掉

PE1

```

1 int g0/0/0
2 undo ipv6 address 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC20/127
3 ipv6 address 2012:: 127

```

PE2

```

1 int g0/0/0
2 undo ipv6 address 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC21/127
3 ipv6 address 2012::1 127

```

此时 ASBR3 查看 就会多出 2012:: 前缀的路由

```

1 [ASBR3]disp bgp ipv6 routing-table
2 Total Number of Routes: 2
3 *> Network   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
4    NextHop   : 2570:CCDD:CCBB:3CAF:EFBE:ACDD:CCDB:5701 LocPrf      :
5
6    MED       :                                         PrefVal   : 0
7
8    Label     :
9    Path/Ogn  : 100 ?
10 *> Network   : 2012::                                         PrefixLen : 127
11    NextHop   : 2570:CCDD:CCBB:3CAF:EFBE:ACDD:CCDB:5701 LocPrf      :
12
13    MED       : 2430                                         PrefVal   : 0
14
15    Label     :
16    Path/Ogn  : 100 ?

```

我们要做的事情呢就简单了，就是在 ASBR1 上把 2012:: 前缀的 过滤 掉

C、（不能使用route-policy）

ASBR1 配置

1. 先写一个 ipv6 prefix 匹配出 DC00/120

```
1 ip ipv6-prefix toASBR3 permit 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 120
```

2. 在 bgp ipv6-family 下对邻居 ASBR3 执行 ipv6 prefix 只往外发送汇总路由

```
1 bgp 100
2   ipv6-family unicast
3     peer 2570:CCDD:CCBB:3CAF:EFBE:ACDD:CCDB:5700 ipv6-prefix toASBR3 export
```

3. 此时 ASBR3 上查看 --- 此时就只有 汇总路由了

```
1 [ASBR3]disp bgp ipv6 routing-table
2
3   Total Number of Routes: 1
4   *> Network   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
5       NextHop   : 2570:CCDD:CCBB:3CAF:EFBE:ACDD:CCDB:5701 LocPrf    :
6
7       MED       :                                         PrefVal   : 0
8
9       Label     :
10      Path/Ogn  : 100 ?
```

D、将ASBR3的loopback0通告入BGP4+

ASBR3 配置

```
1 bgp 200
2   ipv6-family unicast
3     network 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07 128
```

2. PE1,PE2学习到ASBR3 loopback0的BGP4+明细路由（3）

PE1 上查看 ipv6 routing --- 发现只有自己的链路本地地址

```

1 [PE1]disp bgp ipv6 routing-table
2   Total Number of Routes: 2
3     i Network   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
4       NextHop   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf    : 100
5
6       MED       :                                         PrefVal   : 0
7
8       Label     :
9       Path/Ogn  : ?
10    i Network   : 2012::                                         PrefixLen : 127
11      NextHop   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf    : 100
12
13      MED       : 2430                                         PrefVal   : 0
14
15      Label     :
16      Path/Ogn  : ?

```

解法: RR1 和 P1 部署 ipv6 路由渗透

PP1、P1

1. 写一个 `ipv6 prefix` 把所有的环回口匹配出来

```

1 ip ipv6-prefix LOOP permit 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 120
  greater-equal 128 less-equal 128 # 正确
2 ip ipv6-prefix LOOP permit 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC 120 greater-
  equal 128 less-equal 128 # 错误

```

2. 路由渗透时候调用 `filter-policy`

```

1 isis
2   ipv6 import-route isis level-2 into level-1 filter-policy ipv6-prefix LOOP

```

3. PE1、PE2 上查看 --- 发现没有 ASBR3 的明细路由 `DC07`，只有一条汇总路由 + `2012::`

```

1 [PE1]disp bgp ipv6 routing-table
2
3   Total Number of Routes: 2
4   *>i Network   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
5     NextHop   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf    : 100
6
7     MED       :                                         PrefVal   : 0
8
9     Label     :
10    Path/Ogn  : ?
11  *>i Network   : 2012::                                         PrefixLen : 127
12    NextHop   : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf    : 100
13
14    MED       : 2430                                         PrefVal   : 0
15
16    Label     :
17    Path/Ogn  : ?

```

4. 在 RR1、P1 上查看也是如此 --- 这就说明 ASBR3 的明细正好被 ASBR1 的路由汇总汇总了

注意：考场可能出现ASBR3的环回口路由在汇总路由范围内，也被ASBR1抑制明细，需要配置ASBR1不抑制ASBR3的明细路由

解法：ASBR1 写一个 `ipv6 prefix` 把 `DC07` 匹配出来，在写一个 `route-policy deny DC07`，放行其它

然后汇总的时候调用 `suppress-policy` 完事了

ASBR1 配置

A、写一个 `ipv6 prefix` 把 `DC07` 匹配出来

```
1 ip ipv6-prefix ASBR3 permit 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07 128
```

B、写一个 `route-policy`

```
1 route-policy SUPPRESS deny node 10
2   if-match ipv6 address prefix-list ASBR3
3 route-policy SUPPRESS permit node 20
```

C、`aggregate` 时加上 `suppress-policy`

```
1 bgp 100
2   ipv6-family unicast
3     aggregate 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 120 detail-suppressed
      suppress-policy SUPPRESS
```

D、P1、RR1 上查看 --- 发现 `DC07` 没有最优

```
1 [RR1]disp bgp ipv6 routing-table
2
3 Total Number of Routes: 3
4 *>i Network : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
5   NextHop : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf : 100
6
7   MED : PrefVal : 0
8
9   Label :
10  Path/Ogn : ?
11 i Network : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07 PrefixLen : 128
12 NextHop : 2570:CCDD:CCBB:3CAF:EFCE:ACDD:CCDB:5700 LocPrf : 100
13
14 MED : 0 PrefVal : 0
15
16 Label :
17 Path/Ogn : 200 i
18 *>i Network : 2012:: PrefixLen : 127
19 NextHop : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf : 100
20
21 MED : 2430 PrefVal : 0
22
23 Label :
24 Path/Ogn : ?
```

这是怎么回事呢？啊原来是 ASBR1 从 EBGP 学来的路由 通过 IBGP 传递给 RR 时候，要设置 `next-hop-local`

E、ASBR1 配置 `next-hop-local`

```
1  bgp 100
2  ipv6-family unicast
3  peer 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC03 next-hop-local
```

F、PE1、PE2 上查看 -- 发现有汇总也有 ASBR3 的明细路由了

```
1  [PE1]disp bgp ipv6 routing-table
2
3  Total Number of Routes: 3
4  *>i Network : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC00 PrefixLen : 120
5  NextHop : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf : 100
6
7  MED : PrefVal : 0
8
9  Label :
10 Path/Ogn : ?
11 *>i Network : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07 PrefixLen : 128
12 NextHop : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf : 100
13
14 MED : 0 PrefVal : 0
15
16 Label :
17 Path/Ogn : 200 i
18 *>i Network : 2012:: PrefixLen : 127
19 NextHop : 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC05 LocPrf : 100
20
21 MED : 2430 PrefVal : 0
22
23 Label :
24 Path/Ogn : ?
```

PE1 访问 ASBR3

```
1  ping ipv6 -a 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC01
2  2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
3  PING 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07 : 56 data bytes, press CTRL_C
4  to
5  break
6  Reply from 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
7  bytes=56 Sequence=1 hop limit=62 time = 70 ms
8  Reply from 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
9  bytes=56 Sequence=2 hop limit=62 time = 100 ms
10 Reply from 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
11 bytes=56 Sequence=3 hop limit=62 time = 80 ms
12 Reply from 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
13 bytes=56 Sequence=4 hop limit=62 time = 80 ms
14 Reply from 2000:EAD8:99EF:C03E:B2AD:9EFF:32DD:DC07
15 bytes=56 Sequence=5 hop limit=62 time = 80 ms
```

3. 请在PE1使能某特性，以确保PE1在启动过程（从物理接口UP）到各协议邻居建立，PE2-ASBR3的 IPV6 ping无丢包（4）

PE1 配置

A、IGP 与 BGP 联动

```
1 isis
2 set-overload on-startup wait-for-bgp
```

B、IGP 与 IDP 联动

```
1 int Ip-Trunk1
2 isis ldp-sync
3 int g0/0/0
4 isis ldp-sync
```

下一周任务

LAB1 - LAB2 敲熟

论述题

TS: 排错

TAC: 诊断

