

## 知 某局点 S6850 vpn 路由互引问题

BGP 柯辉 2022-05-25 发表

### 组网及说明

两台6850 互连, 每台设备是各有两个 vpn 实例, 即 vpn instance IN 和 OUT, 两个实例内各自相互建立 BGP 邻居

告警信息

暂无

## 问题描述

现场路由本地无法互相引入。

SW2的BGP配置为：

```
<SW02>dis c c bgp
bgp 65211
ip vpn-instance In
peer 10.10.101.186 as-number 65211
address-family ipv4 unicast
balance 16
network 10.10.100.157 255.255.255.255
peer 10.10.101.186 enable
```

```
ip vpn-instance Out
peer 10.10.101.154 as-number 65211
address-family ipv4 unicast
balance 16
peer 10.10.101.154 enable
```

本都 两个 vpn 通过RT值打通

```
ip vpn-instance In
route-distinguisher 203:1
#
address-family ipv4
vpn-target 103:1 import-extcommunity
vpn-target 103:1 export-extcommunity
#
ip vpn-instance Out
route-distinguisher 103:1
#
address-family ipv4
vpn-target 103:1 import-extcommunity
vpn-target 103:1 export-extcommunity
#
```

SW1 的BGP配置为：

```
<SW01>dis c c bgp
bgp 65211
ip vpn-instance In
peer 10.10.101.185 as-number 65211
address-family ipv4 unicast
balance 16
peer 10.10.101.185 enable
```

```
ip vpn-instance Out
peer 10.10.101.153 as-number 65211
address-family ipv4 unicast
balance 16
peer 10.10.101.153 enable
```

本都 两个 vpn 通过RT值打通

```
ip vpn-instance In
route-distinguisher 203:1
#
address-family ipv4
vpn-target 103:1 import-extcommunity
vpn-target 103:1 export-extcommunity
#
ip vpn-instance Out
route-distinguisher 103:1
#
address-family ipv4
```

```
vpn-target 103:1 import-extcommunity
```

```
vpn-target 103:1 export-extcommunity
```

#### 过程分析

当前SW2的 vpn instance In, network 了一条路由 network 10.10.100.157 255.255.255.255

SW2的本地RT值打通了, 所以vpn instance In和Out都会有一条 10.10.100.157 的路由:

SW2的 ip vpn instance In:

```
<SW02>dis ip routing-table vpn-instance In
```

```
Destination/Mask Proto Pre Cost NextHop Interface
```

```
10.10.100.157/32 Direct 0 0 127.0.0.1 InLoop0
```

ip vpn instance Out:

```
<AGG-SW02>dis ip routing-table vpn-instance Out
```

```
Destinations : 13 Routes : 13 Destination/Mask Proto Pre Cost NextHop Interface
```

```
10.10.100.157/32 BGP 100 0 127.0.0.1 InLoop0 // vpn 实例Out 通过BGP 学到了本地 vpn In 学到了 10.10.100.157 的路由
```

对于SW1 通过 BGP学到该路由:

SW1的 vpn instance In :

```
<SW01>dis ip routing-table vpn-instance In
```

```
Destination/Mask Proto Pre Cost NextHop Interface
```

```
10.10.100.157/32 BGP 100 0 10.10.101.155 Vlan103  
10.10.101.187 Vlan203
```

SW1的 vpn instance Out:

```
<SW01>dis ip routing-table vpn-instance Out
```

```
Destination/Mask Proto Pre Cost NextHop Interface
```

```
10.10.100.157/32 BGP 100 0 10.10.101.155 Vlan103
```

```
interface Vlan-interface103
```

```
ip binding vpn-instance Out
```

```
ip address 10.10.101.154 255.255.255.248
```

```
interface Vlan-interface203
```

```
ip binding vpn-instance In
```

```
ip address 10.10.101.186 255.255.255.248
```

可以看到 在SW1 的Out的 vpn中 路由表在, 只有对端SW2 Out 实例传递过来的 路由; SW1的 In的vpn 路由表中, 除了对端SW2的 In vpn实例传递过来的外, 还有从SW1的Oot vpn实例同步过的, 而SW1的In vpn 实例的路由没有同步到 SW1的Out 实例中

## 解决方法

排查发现，本地的两个vpn 实例内的路由，同步vpn实例同步的路由 在自己 vpn 中也与去往相同目的地址的路由时，会存在时序，即当前SW1的 Out vpn 实例的去往10.10.100.157的路由同步给了本地vpn 实例 In，而 vpn 实例 In也有一条去往 10.10.100.57 的路由了，这个时候，In实例收到Out同步的后，不会在将自己的路由同步给本地的vpn 实例Out，配置命令 vpn-route cross multipath 解决这个问题

在SW1的两个vpn 实例的BGP，ipv4地址族下配置 vpn-route cross multipath，解决

```
address-family ipv4 unicast
```

```
vpn-route cross multipath
```

