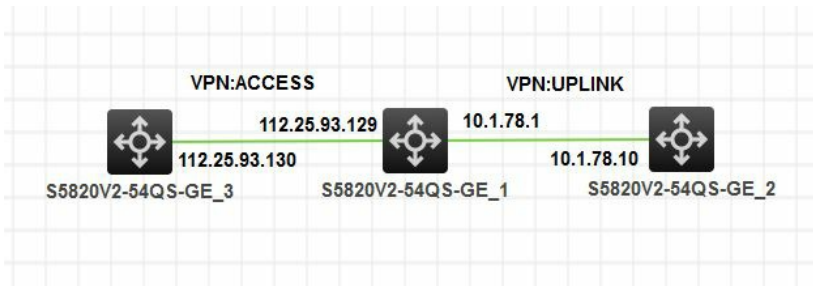


一、组网环境:



二、组网说明

中间98设备上存在两个VPN实例ACCESS和UPLINK，现在需要通过路由互引和路由迭代以及PBR的方式实现两边互通。

三、实现方式

1、直接通过缺省路由互引达到互通。

在中间设备上通过增加如下的静态路由来达到两vpn之间的互通。

```
ip route-static vpn-instance UPLINK 112.25.93.0 24 vpn-instance ACCESS 112.25.93.130
ip route-static vpn-instance ACCESS 10.1.78.0 24 vpn-instance UPLINK 10.1.78.10
```

此时两个vpn中均有到对方的路由，ping此时可通：

```
[H3C]ping 10.1.78.10
Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
56 bytes from 10.1.78.10: icmp_seq=0 ttl=254 time=1.944 ms
56 bytes from 10.1.78.10: icmp_seq=1 ttl=254 time=1.556 ms
56 bytes from 10.1.78.10: icmp_seq=2 ttl=254 time=1.689 ms
56 bytes from 10.1.78.10: icmp_seq=3 ttl=254 time=1.619 ms
56 bytes from 10.1.78.10: icmp_seq=4 ttl=254 time=1.404 ms
--- Ping statistics for 10.1.78.10 ---
```

2、通过路由迭代的方式来实现，即ACCESS中的流量匹配缺省路由扔到一个设备上UPLINK中的一个不存在的地址1.1.1.3，之后该不存在的地址通过明细路由迭代到出接口，而从UPLINK中回来的流量通过匹配明细静态路由到达ACCESS。

添加路由如下：

```
ip route-static vpn-instance ACCESS 0.0.0.0 0 vpn-instance UPLINK 1.1.1.3
ip route-static vpn-instance UPLINK 1.1.1.3 32 10.1.78.10
ip route-static vpn-instance UPLINK 112.25.93.0 24 vpn-instance ACCESS 112.25.93.130
```

查看路由

```
[H3C]dis ip routing-table vpn-instance ACCESS
```

Destinations : 13 Routes : 13

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/0	Static	60	0	1.1.1.3	Vlan78
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
112.25.93.128/26	Direct	0	0	112.25.93.129	Vlan10
112.25.93.128/32	Direct	0	0	112.25.93.129	Vlan10
112.25.93.129/32	Direct	0	0	127.0.0.1	InLoop0
112.25.93.191/32	Direct	0	0	112.25.93.129	Vlan10
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0
224.0.0.0/4	Direct	0	0	0.0.0.0	NULL0
224.0.0.0/24	Direct	0	0	0.0.0.0	NULL0
255.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

```
[H3C]dis ip routing-table vpn-instance UPLINK
```

Destinations : 15 Routes : 15

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0

```

1.1.1.3/32      Static 60 0      10.1.78.10  Vlan78
10.1.78.0/24   Direct 0 0      10.1.78.1   Vlan78
10.1.78.0/32   Direct 0 0      10.1.78.1   Vlan78
10.1.78.1/32   Direct 0 0      127.0.0.1   InLoop0
10.1.78.255/32 Direct 0 0      10.1.78.1   Vlan78
112.25.93.0/24 Static 60 0      112.25.93.130 Vlan10
127.0.0.0/8    Direct 0 0      127.0.0.1   InLoop0
127.0.0.0/32   Direct 0 0      127.0.0.1   InLoop0
127.0.0.1/32   Direct 0 0      127.0.0.1   InLoop0
127.255.255.255/32 Direct 0 0      127.0.0.1   InLoop0
224.0.0.0/4    Direct 0 0      0.0.0.0     NULL0
224.0.0.0/24   Direct 0 0      0.0.0.0     NULL0
255.255.255.255/32 Direct 0 0      127.0.0.1   InLoop0

```

ping测试可以互通:

```
[H3C]ping 10.1.78.10
```

```

Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
56 bytes from 10.1.78.10: icmp_seq=0 ttl=254 time=2.246 ms
56 bytes from 10.1.78.10: icmp_seq=1 ttl=254 time=1.456 ms
56 bytes from 10.1.78.10: icmp_seq=2 ttl=254 time=1.733 ms
56 bytes from 10.1.78.10: icmp_seq=3 ttl=254 time=1.615 ms
56 bytes from 10.1.78.10: icmp_seq=4 ttl=254 time=1.426 ms

```

```
--- Ping statistics for 10.1.78.10 ---
```

```

5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 1.426/1.695/2.246/0.297 ms

```

3、通过PBR的方式达到互通，在2的基础上删除ACCESS中的缺省路由，只有通过PBR将ACCESS中的流量匹配到UPLINK中的一个不存在的地址1.1.1.3，之后通过迭代找到出接口。如下:

```

undo ip route-static vpn-instance ACCESS 0.0.0.0 0 vpn-instance UPLINK 1.1.1.3
#
acl number 2900 name SR_IPLIST
rule 5 permit source 112.25.93.130 0
#
[H3C]dis ip policy-based-route
Policy name: h3c
node 10 permit:
if-match acl 2900
apply next-hop vpn-instance UPLINK 1.1.1.3

```

将pbr应用在ACCESS的vlan虚接口下，此时ACCESS中的路由如下:

```

[H3C]DIS IP routing-table vpn-instance ACCESS
Destinations : 12   Routes : 12
Destination/Mask  Proto Pre Cost   NextHop      Interface
0.0.0.0/32        Direct 0 0      127.0.0.1    InLoop0
112.25.93.128/26  Direct 0 0      112.25.93.129 Vlan10
112.25.93.128/32  Direct 0 0      112.25.93.129 Vlan10
112.25.93.129/32  Direct 0 0      127.0.0.1    InLoop0
112.25.93.191/32  Direct 0 0      112.25.93.129 Vlan10
127.0.0.0/8       Direct 0 0      127.0.0.1    InLoop0
127.0.0.0/32      Direct 0 0      127.0.0.1    InLoop0
127.0.0.1/32      Direct 0 0      127.0.0.1    InLoop0
127.255.255.255/32 Direct 0 0      127.0.0.1    InLoop0
224.0.0.0/4       Direct 0 0      0.0.0.0      NULL0
224.0.0.0/24      Direct 0 0      0.0.0.0      NULL0
255.255.255.255/32 Direct 0 0      127.0.0.1    InLoop0

```

此时由于PBR的原因ACCESS中的流量到达了UPLINK中的1.1.1.3，之后迭代到出接口，ping测试可以通:

```
[H3C]ping 10.1.78.10
```

```

Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
56 bytes from 10.1.78.10: icmp_seq=0 ttl=254 time=1.944 ms
56 bytes from 10.1.78.10: icmp_seq=1 ttl=254 time=1.556 ms
56 bytes from 10.1.78.10: icmp_seq=2 ttl=254 time=1.689 ms
56 bytes from 10.1.78.10: icmp_seq=3 ttl=254 time=1.619 ms
56 bytes from 10.1.78.10: icmp_seq=4 ttl=254 time=1.404 ms

```

4、注意通过3中的方式PBR到UPLINK中不存在的地址没有明细路由的话是不会通的，即使存在缺省路由，如ACCESS中匹配PBR到UPLINK中的1.1.1.1，UPLINK中没有1.1.1.1的明细出口路由，只有缺省路由指向1.1.1.3，如下的配置：

```
#
acl number 2900 name SR_IPLIST
rule 5 permit source 112.25.93.130 0
#
[H3C]dis ip policy-based-route
Policy name: h3c
node 10 permit:
if-match acl 2900
apply next-hop vpn-instance UPLINK 1.1.1.1
#
ip route-static vpn-instance UPLINK 0.0.0.0 0 1.1.1.3
ip route-static vpn-instance UPLINK 1.1.1.3 32 10.1.78.10
ip route-static vpn-instance UPLINK 112.25.93.0 24 vpn-instance ACCESS 112.25.93.130
```

此时UPLINK中的路由如下：

```
[H3C]dis ip routing-table vpn-instance UPLINK
Destinations : 15 Routes : 15
Destination/Mask Proto Pre Cost NextHop Interface
0.0.0.0/0 Static 60 0 1.1.1.3 Vlan78
0.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
1.1.1.3/32 Static 60 0 10.1.78.10 Vlan78
10.1.78.0/24 Direct 0 0 10.1.78.1 Vlan78
10.1.78.0/32 Direct 0 0 10.1.78.1 Vlan78
10.1.78.1/32 Direct 0 0 127.0.0.1 InLoop0
10.1.78.255/32 Direct 0 0 10.1.78.1 Vlan78
112.25.93.0/24 Static 60 0 112.25.93.130 Vlan10
127.0.0.0/8 Direct 0 0 127.0.0.1 InLoop0
127.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
127.0.0.1/32 Direct 0 0 127.0.0.1 InLoop0
127.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
224.0.0.0/4 Direct 0 0 0.0.0.0 NULL0
224.0.0.0/24 Direct 0 0 0.0.0.0 NULL0
255.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
```

此时进行ping测试是不通的：

```
[H3C]ping 10.1.78.10
Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
Request time out
Request time out
Request time out
Request time out
Request time out
```

--- Ping statistics for 10.1.78.10 ---

5、由于UPLINK中没有1.1.1.1的明细路由，即使存在缺省路由也是不能达到两边的互通，此时在UPLINK中添加1.1.1.1的明细路由进行验证，

```
ip route-static vpn-instance UPLINK 1.1.1.1 32 10.1.78.10

[H3C]dis ip routing-table vpn-instance UPLINK
Destinations : 16 Routes : 16
Destination/Mask Proto Pre Cost NextHop Interface
0.0.0.0/0 Static 60 0 1.1.1.3 Vlan78
0.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
1.1.1.1/32 Static 60 0 10.1.78.10 Vlan78
1.1.1.3/32 Static 60 0 10.1.78.10 Vlan78
10.1.78.0/24 Direct 0 0 10.1.78.1 Vlan78
10.1.78.0/32 Direct 0 0 10.1.78.1 Vlan78
10.1.78.1/32 Direct 0 0 127.0.0.1 InLoop0
10.1.78.255/32 Direct 0 0 10.1.78.1 Vlan78
112.25.93.0/24 Static 60 0 112.25.93.130 Vlan10
127.0.0.0/8 Direct 0 0 127.0.0.1 InLoop0
```

```
127.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
127.0.0.1/32 Direct 0 0 127.0.0.1 InLoop0
127.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
224.0.0.0/4 Direct 0 0 0.0.0.0 NULL0
224.0.0.0/24 Direct 0 0 0.0.0.0 NULL0
255.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
```

可以 ping 通:

```
[H3C]ping 10.1.78.10
Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
56 bytes from 10.1.78.10: icmp_seq=0 ttl=254 time=2.871 ms
56 bytes from 10.1.78.10: icmp_seq=1 ttl=254 time=2.205 ms
56 bytes from 10.1.78.10: icmp_seq=2 ttl=254 time=2.086 ms
56 bytes from 10.1.78.10: icmp_seq=3 ttl=254 time=2.435 ms
56 bytes from 10.1.78.10: icmp_seq=4 ttl=254 time=2.204 ms
```

--- Ping statistics for 10.1.78.10 ---

6、或者通过明细路由进行迭代来实现互通，在UPLINK中添加一条明细路由指向1.1.1.3,，如下:

```
undo ip route-static vpn-instance UPLINK 1.1.1.1 32 10.1.78.10
ip route-static vpn-instance UPLINK 1.1.1.1 32 1.1.1.3
```

```
[H3C]dis ip routing-table vpn-instance UPLINK
Destinations : 16 Routes : 16
Destination/Mask Proto Pre Cost NextHop Interface
0.0.0.0/0 Static 60 0 1.1.1.3 Vlan78
0.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
1.1.1.1/32 Static 60 0 1.1.1.3 Vlan78
1.1.1.3/32 Static 60 0 10.1.78.10 Vlan78
10.1.78.0/24 Direct 0 0 10.1.78.1 Vlan78
10.1.78.0/32 Direct 0 0 10.1.78.1 Vlan78
10.1.78.1/32 Direct 0 0 127.0.0.1 InLoop0
10.1.78.255/32 Direct 0 0 10.1.78.1 Vlan78
112.25.93.0/24 Static 60 0 112.25.93.130 Vlan10
127.0.0.0/8 Direct 0 0 127.0.0.1 InLoop0
127.0.0.0/32 Direct 0 0 127.0.0.1 InLoop0
127.0.0.1/32 Direct 0 0 127.0.0.1 InLoop0
127.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
224.0.0.0/4 Direct 0 0 0.0.0.0 NULL0
224.0.0.0/24 Direct 0 0 0.0.0.0 NULL0
255.255.255.255/32 Direct 0 0 127.0.0.1 InLoop0
```

[H3C]

[H3C]

```
[H3C]ping 10.1.78.10
Ping 10.1.78.10 (10.1.78.10): 56 data bytes, press CTRL_C to break
56 bytes from 10.1.78.10: icmp_seq=0 ttl=254 time=2.647 ms
56 bytes from 10.1.78.10: icmp_seq=1 ttl=254 time=2.103 ms
56 bytes from 10.1.78.10: icmp_seq=2 ttl=254 time=2.319 ms
56 bytes from 10.1.78.10: icmp_seq=3 ttl=254 time=2.172 ms
56 bytes from 10.1.78.10: icmp_seq=4 ttl=254 time=2.291 ms
```

--- Ping statistics for 10.1.78.10 ---

四、总结分析

两个vpn之间可以通过路由互引的方式达到互通，也可以通过PBR方式将流量引向别的VPN，当进行路由迭代时一定要确保VPN中具有明细路由，否则不能迭代。