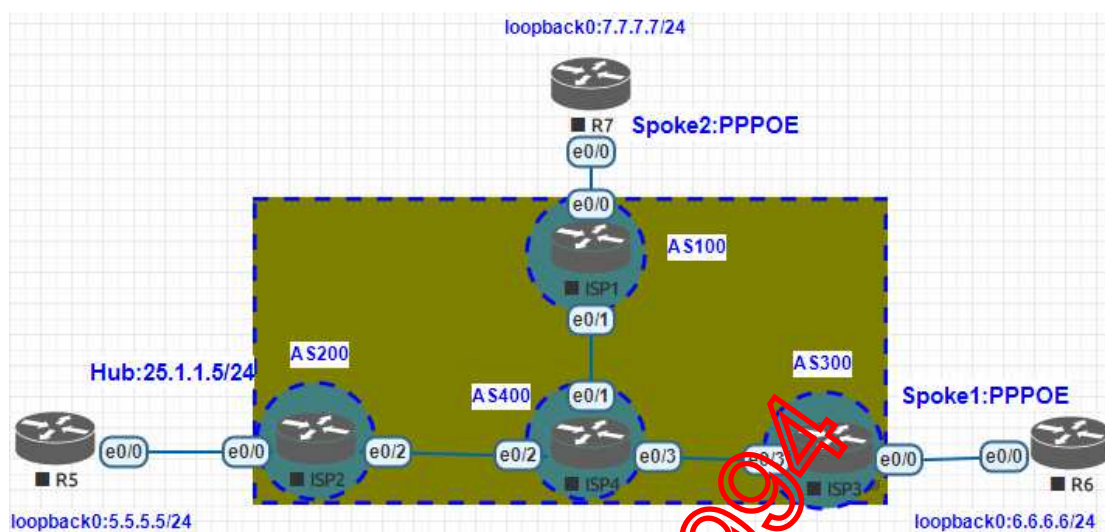


带 NAT、PPPOE 的 DMVPN

一、拓扑



要求:

- 1.R5、R6、R7 分别模拟某企业的三个场点的边界路由器，其中 R5 为 HUB 的边界路由器，R6、R7 分别为 SPOKE1、SPOKE2 处的边界路由器。
- 2.ISP1、ISP2、ISP3、ISP4 模拟 ISP 网络中的设备，ISP4 上有环回口 4.4.4.4/24，模拟 Internet 上的网站。
- 3.R5 通过公网地址 25.1.1.5/24 接入 Internet，R6、R7 分别通过 PPPOE 方式连入 Internet。
- 4.R5、R6、R7 上的环回口分别模拟企业三个场点的私网地址，并且在 R5、R6、R7 上配置 DMVPN，要求企业私网之间互访时，数据流进行加密；私网访问 Internet 时，不加密。

二、配置

1.IP 地址和 BGP 配置 (省略)。

通过在 ISP1、ISP2、ISP3、ISP4 上配置 EBGP，并且通过重发布的方式将直连路由通告进 BGP 中，使 ISP1、ISP2、ISP3 可正常访问 ISP4 上的 4.4.4.4/24。

```
ISP1#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
ISP1#
```

可以看到 ISP1、ISP2、ISP3 正常访问 ISP4 的环回口，说明 ISP 网络配置成功。

2.HUB 和 SPOKE1、SPOKE2 接入 Internet

(1)HUB 上的配置

HUB，即中心站点的边界路由器（图中为 R5），必须采用公网 IP 地址，即拥有静态的公网 IP 地址接入 Internet，不能通过动态 IP 地址接入 Internet，这是因为 MGRE 的配置中，SPOKE 的 IP 地址一般是动态的，为了在 Internet 随时找到 SPOKE 的公网地址，必须通过 NHRP 表进行映射和维持，所以要求 HUB 节点必须要有公网地址。

```

R5(config)#interface e0/0
R5(config-if)#ip address 25.1.1.5 255.255.255.0
R5(config-if)#no shutdown
R5(config-if)#exit
R5(config)#ip
*Jan 29 09:54:36.593: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state
*Jan 29 09:54:37.593: %LINEPROTO-5-UPDOWN: Line protocol on Interface Eth
to up
R5(config)#ip route 0.0.0.0 0.0.0.0 e0/0 10
%Default route without gateway, it not a point-to-point interface, may imp

```

由于 R5 通过公网 IP 地址接入 Internet，所以 R5 上的配置很简单。

```

R5#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
R5#ping 4.4.4.4 source 5.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
.....
Success rate is 0 percent (0/5)
R5#

```

此时可以看到 R5 能够正常访问 Internet，但 R5 背后的私网没法访问 Internet，这是因为 Internet 不可能拥有去往 R5 背后私网的路由，数据包没法返回，所以访问不通。

解决方法有两种：①做 NAT②在 Internet 上写私网路由（不可行），但此处我们暂不解决这个问题，在后面解决。

(2)SPOKE1 上的配置

SPOKE1，即分支站点 1 的边界路由器，即图中的 R6，通过 PPPOE 方式接入 Internet。ISP3 即为 PPPOE 的服务端，通过 DHCP 的方式为 R6 动态分配 IP 地址，R6 为 PPPOE 的客户端，通过 PPPOE 连接到 Internet 中，并且通过 DHCP 方式动态地从 ISP3 处获得 IP 地址。

①PPPOE 客户端配置—R6 配置

```

R6(config)#interface e0/0
R6(config-if)#pppoe enable
R6(config-if)#pppoe-client dial-pool-number 1
R6(config-if)#no shutdown

```

先在物理接口上开启 PPPOE 功能，并将 dialer 接口和物理接口关联。

```

R6(config)#interface dialer 1
R6(config-if)#encapsulation ppp
R6(config-if)#ppp chap hostname 01012345
R6(config-if)#ppp chap password 123456

```

再配置 dialer 接口的封装方式，验证方式。

```

R6(config-if)#dialer pool 1
R6(config-if)#mtu 1492
R6(config-if)#ip address dhcp
R6(config-if)#no shutdown

```

最后定义 dialer 口的 IP 地址获取方式和 MTU 大小。

②PPPOE 服务器端配置—ISP3 配置

```

ISP3(config)#username 01012345 password 12345
ISP3(config)#ip dhcp pool PPPOE
ISP3(dhcp-config)#network 36.1.1.0 /24
ISP3(dhcp-config)#default-router 36.1.1.3
ISP3(dhcp-config)#exit
ISP3(config)#ip dhcp excluded-address 36.1.1.1 36.1.1.3

```

先配置本地帐号和 DHCP 地址池，本地帐号用于 PPPOE 拨号认证，而 DHCP 地址池用于 PPPOE 认证通过后，给 PPPOE 客户端动态分配 IP 地址，其中地址池中的网关为 36.1.1.3，

该地址不是 ISP3 的物理接口 e0/0 的 IP 地址，而是下面要配置的虚模板的 IP 地址。

```
ISP3(config)#interface virtual-template 1
ISP3(config-if)#encapsulation ppp
*Jan 29 10:21:33.079: %LINK-3-UPDOWN: Interface Virtual-Template1, changed state to
ISP3(config-if)#encapsulation ppp
ISP3(config-if)#ppp authentication chap
ISP3(config-if)#mtu 1492
ISP3(config-if)#ip address 36.1.1.3 255.255.255.0
ISP3(config-if)#peer default ip address dhcp-pool PPPOE
ISP3(config-if)#no shutdown
```

配置虚模板 virtual-template1，其 IP 地址为 DHCP 地址池的网关—36.1.1.3/24，并且定义其对端的 IP 地址从 DHCP 地址池 PPPOE 中获取。

```
ISP3(config)#bba-group pppoe cisco
ISP3(config-bba-group)#
*Jan 29 10:23:50.158: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Template1, changed state to up
ISP3(config-bba-group)#virtual-template 1
*Jan 29 10:23:50.158: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up
ISP3(config-bba-group)#exit
ISP3(config)#
```

定义 bba 组，其名称为 cisco，调用虚板 virtual-template1。

```
ISP3(config)#interface e0/0
ISP3(config-if)#pppoe enable group cisco
ISP3(config-if)#no shutdown
```

此时 PPPOE 服务器端和 PPPOE 客户端配置完成。

在这里我们强调：

- ①配置 PPPOE 时，先配置 PPPOE 客户端，再配置 PPPOE 服务器端，否则可能起不来。
- ②配置 PPPOE 服务器端时，按帐号和地址池、虚模板、BBA 组、物理接口调用的顺序来配置。
- ③配置 PPPOE 客户端时，按物理接口、Dialer 接口的顺序配置。

当 PPPOE 配置成功后，还必须在 PPPOE 客户端上写一条默认路由，其出接口不是 SPOKE1 的物理接口，而是物理接口上调用的 dialer 接口

```
R6(config)#ip route 0.0.0.0 0.0.0.0 dialer 1 10
R6(config)#
```

此时，在 SPOKE1 上测试，

```
R6#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R6#ping 4.4.4.4 source 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 6.6.6.6
.....
Success rate is 0 percent (0/5)
R6#
```

可以看到 SPOKE1 可以正常访问 Internet，但 SPOKE1 背后的私网不能访问 Internet，原因同样是没有回程路由。

(3)SPOKE2 上的配置

SPOKE2，即 R7，同样通过 PPPOE 的方式接入 Internet，所以 ISP1 就为 PPPOE 服务器端，R7 就为 PPPOE 客户端，配置如下：

①PPPOE 客户端配置—R7 配置

```

R7(config)#interface e0/0
R7(config-if)#pppoe enable
R7(config-if)#
*Jan 29 10:40:36.120: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
R7(config-if)#ppp
*Jan 29 10:40:36.120: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
R7(config-if)#pppoe-client dial-pool-number 1
R7(config-if)#no shutdown

```

```

R7(config)#interface dialer 1
R7(config-if)#encapsulation ppp
R7(config-if)#ppp chap hostname 02012345
R7(config-if)#ppp chap password 123456
R7(config-if)#dial pool 1
R7(config-if)#mtu 1492
R7(config-if)#ip address negotiate
R7(config-if)#no shutdown

```

注意此处，PPPOE 客户端是通过协商的方式获得 IP 地址，而不是 DHCP 方式。

②PPPOE 服务端配置—ISP1 配置

```

ISP1(config)#username 02012345 password 123456
ISP1(config)#ip local pool PPPOE 17.1.1.10 17.1.1.150
ISP1(config)#
ISP1(config)#interface virtual-template 1
ISP1(config-if)#encapsulation ppp
ISP1(config-if)#ppp authentication chap
ISP1(config-if)#mtu 1492
ISP1(config-if)#ip address 17.1.1.1 255.255.255.0
ISP1(config-if)#peer default ip address pool PPPOE
ISP1(config-if)#no shutdown

```

服务端定义客户端的 IP 地址从本地的地址池 (PPPOE) 中获得，不是通过 DHCP 服务器的方式动态获取。

```

ISP1(config)#bba-group pppoe cisco
ISP1(config-bba-group)#v
*Jan 29 10:50:09.689: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state to up
ISP1(config-bba-group)#virtual-template 1
*Jan 29 10:50:09.689: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up
ISP1(config-bba-group)#virtual-template 1
ISP1(config)#interface e0/0
ISP1(config-if)#pppoe enable group cisco
ISP1(config-if)#no shutdown

```

③PPPOE 客户端上配默认路由

```

R7(config)#ip route 0.0.0.0 0.0.0.0 dialer 1 10
R7(config)#

```

同样，SPOKE2 能正常访问 Internet，但其背后的私网不能访问 Internet。

```

R7#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R7#ping 4.4.4.4 source 7.7.7.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 7.7.7.7
.....
Success rate is 0 percent (0/5)

```

3.MGRE 和 NRHP 配置

GRE 默认为点对点模式，但在 DMVPN 中必须为多点 GRE，现在 HUB 和 SPOKE1、SPOKE2 上配置多点 GRE，并且多点 GRE 的 IP 地址分别为 100.1.1.5/24、100.1.1.6/24、100.1.1.7/24。

(1)MGRE 配置

①HUB 上的配置

```
R5(config)#interface tunnel 567
R5(config-if)#tunnel mode gre
*Jan 29 10:59:04.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o down
R5(config-if)#tunnel mode gre multipoint
R5(config-if)#tunnel source 25.1.1.5
R5(config-if)#ip address
*Jan 29 10:59:19.054: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o up
R5(config-if)#ip address 100.1.1.5 255.255.255.0
```

②SPOKE1 上的配置

```
R6(config)#interface tunnel 567
R6(config-if)#tunnel mode
*Jan 29 11:00:36.326: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o down
R6(config-if)#tunnel mode gre multipoint
R6(config-if)#tunnel source dialer 1
R6(config-if)#ip address
*Jan 29 11:00:49.642: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o up
R6(config-if)#ip address 100.1.1.6 255.255.255.0
```

此处，SPOKE1 上的 tunnel 源地址为 dialer1 接口，因为 dialer1 上的 IP 地址不是固定的，所以只能采用接口的方式，这一点和 HUB 上的配置不同。

③SPOKE2 上的配置

```
R7(config)#interface tunnel 567
R7(config-if)#tunnel mode gre
*Jan 29 11:02:48.813: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o down
R7(config-if)#tunnel mode gre multipoint
R7(config-if)#tunnel source dialer 1
R7(config-if)#ip address
*Jan 29 11:02:59.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel567,
o up
R7(config-if)#ip address 100.1.1.7 255.255.255.0
```

此时，HUB、SPOKE1、SPOKE2 上的 GRE 配置完成，能否互通？我们测试下

```
R5#ping 100.1.1.6 source 100.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.1.1.6, timeout is 2 seconds:
Packet sent with a source address of 100.1.1.5
.....
Success rate is 0 percent (0/5)
R5#
```

可以发现 GRE 并不能通信，为什么这样？这是因为虽然知道了 GRE 源端，在发起连接时，不知道 GRE 目的端的公网地址的缘故。因此，必须告知 GRE 源端，GRE 目的地端的公网 IP 是什么，这个工作就需要 NHRP 协议来完成。

(2)NHRP 配置

①HUB 配置

```
R5(config)#interface tunnel 567
R5(config-if)#ip nhrp network-id 10
R5(config-if)#ip nhrp authentication cisco
R5(config-if)#ip nhrp map multicast dynamic
R5(config-if)#
```

②SPOKE1 配置

```

R6(config)#interface tunnel 567
R6(config-if)#ip nhrp network-id 10
R6(config-if)#ip nhrp authentication cisco
R6(config-if)#ip nhrp map 100.1.1.5 25.1.1.5
R6(config-if)#ip nhrp map multicast 25.1.1.5
R6(config-if)#ip nhrp nhs 100.1.1.5

```

在 SPOKE1 端，将 HUB 端的公网 IP 地址和 Tunnel 地址进行绑定，并且告知 SPOKE1 接收源地址为 25.1.1.5 的组播报文，这样，SPOKE1 会自动地向 HUB 端进行注册，将自己的 tunnel 地址和公网 IP 地址注册到 HUB 端上，亦即 NHS 处。

③SPOKE2 配置

```

R7(config)#interface tunnel 567
R7(config-if)#ip nhrp network-id 10
R7(config-if)#ip nhrp authentication cisco
R7(config-if)#ip nhrp map 100.1.1.5 25.1.1.5
R7(config-if)#ip nhrp map multicast 25.1.1.5
R7(config-if)#ip nhrp nhs 100.1.1.5

```

配置完成后，测试如下

```

R5#ping 100.1.1.6 source 100.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.1.1.6, timeout is 2 seconds:
Packet sent with a source address of 100.1.1.5
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/3/5 ms
R5#ping 100.1.1.7 source 100.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.1.1.7, timeout is 2 seconds:
Packet sent with a source address of 100.1.1.5
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
R5#

```

可以看到此时 Tunnel 互通。

4.配置路由协议，让 HUB、SPOKE1、SPOKE2 背后的私网互通

此时 HUB、SPOKE1、SPOKE2 可以通过 tunnel 互通，那么就在逻辑上相当于直连，这样我们通过配置动态路由协议，可以让它们背后的私网互通。

动态路由协议，可以选择 Eigrp、OSPF、RIP 等，这里我们根据现实特点，选择 OSPF 协议。

(1)HUB、SPOKE1、SPOKE2 配置 OSPF

①HUB 配置

```

R5(config)#router ospf 110
R5(config-router)#router-id 5.5.5.5
R5(config-router)#exit
R5(config)#interface loopback 0
R5(config-if)#ip ospf 110 area 0
R5(config-if)#exit
R5(config)#interface tunnel 567
R5(config-if)#ip ospf 110 area 0

```

②SPOKE1 配置

```

R6(config)#router ospf 110
R6(config-router)#router-id 6.6.6.6
R6(config-router)#exit
R6(config)#interface loopback 0
R6(config-if)#ip ospf 110 area 0
R6(config-if)#exit
R6(config)#interface tunnel 567
R6(config-if)#ip ospf priority 0
R6(config-if)#ip ospf 110 area 0

```

在 SPOKE1 上，通过调整接口的 ospf 优先级为 0，使 SPOKE1 失去竞选 dr 的机会，从而使 HUB 成为 dr。

③SPOKE2 上配置

```
R7(config)#router ospf 110
R7(config-router)#router-id 7.7.7.7
R7(config-router)#exit
R7(config)#interface loopback 0
R7(config-if)#ip ospf 110 area 0
R7(config-if)#exit
R7(config)#interface tunnel 567
R7(config-if)#ip ospf priority 0
R7(config-if)#ip ospf 110 area 0
R7>
```

此时，发现 OSPF 邻居不停地翻滚，这里由于 OSPF 在 Tunnel 中默认为点对点模式，所以要在 HUB、SPOKE1、SPOKE2 上修改为点对多点或者广播型。

(2)修改 OSPF 为点对多点形式

①HUB 上调为点对多点

```
R5(config)#interface tunnel 567
R5(config-if)#ip ospf network point-to-multipoint
```

②SPOKE1 调为点对多点

```
R6(config)#interface tunnel 567
R6(config-if)#ip ospf network point-to-multipoint
R6(config-if)#exit
```

③SPOKE2 调为点对多点

```
R7(config)#interface tunnel 567
R7(config-if)#ip ospf network point-to-multipoint
```

配置完成后，分别在 HUB、SPOKE1、SPOKE2 上查看路由表及 OSPF 邻居

```
R5#show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address        Interface
7.7.7.7          0    FULL/ -         00:01:43    100.1.1.7      Tunnel567
6.6.6.6          0    FULL/ -         00:01:49    100.1.1.6      Tunnel567
```

```
R5#
6.0.0.0/24 is subnetted, 1 subnets
O       6.6.6.0 [110/1001] via 100.1.1.6, 00:02:02, Tunnel567
7.0.0.0/24 is subnetted, 1 subnets
O       7.7.7.0 [110/1001] via 100.1.1.7, 00:00:35, Tunnel567
100.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       100.1.1.6/32 [110/1000] via 100.1.1.6, 00:03:28, Tunnel567
O       100.1.1.7/32 [110/1000] via 100.1.1.7, 00:00:45, Tunnel567
R5#
```

可以看到 HUB 拥有两个 OSPF 邻居，并且拥有去往 SPOKE1、SPOKE2 的私网路由，下一跳为 SPOKE1、SPOKE2 的 tunnel 接口地址。

```
R6#show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address        Interface
5.5.5.5          0    FULL/ -         00:01:44    100.1.1.5      Tunnel567
R6#
```

```
R6#
5.0.0.0/24 is subnetted, 1 subnets
O       5.5.5.0 [110/1001] via 100.1.1.5, 00:04:43, Tunnel567
7.0.0.0/24 is subnetted, 1 subnets
O       7.7.7.0 [110/2001] via 100.1.1.5, 00:02:51, Tunnel567
100.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       100.1.1.5/32 [110/1000] via 100.1.1.5, 00:05:44, Tunnel567
O       100.1.1.7/32 [110/2000] via 100.1.1.5, 00:02:51, Tunnel567
R6#
```

SPOKE1 仅仅与 HUB 建立了邻居，并且其去往 SPOKE2 的私网路由的下一跳也为 HUB 的 tunnel 地址。

```
R7#show ip ospf neighbor
Neighbor ID      Pri   State   Dead Time   Address        Interface
5.5.5.5          0    FULL/  -         00:01:55     100.1.1.5     Tunnel567
R7#

5.0.0.0/24 is subnetted, 1 subnets
O        5.5.5.0 [110/1001] via 100.1.1.5, 00:01:48, Tunnel567
6.0.0.0/24 is subnetted, 1 subnets
O        6.6.6.0 [110/2001] via 100.1.1.5, 00:01:48, Tunnel567
100.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O        100.1.1.5/32 [110/1000] via 100.1.1.5, 00:01:48, Tunnel567
O        100.1.1.6/32 [110/2000] via 100.1.1.5, 00:01:48, Tunnel567
R7#
```

SPOKE2 同样。

测试 SPOKE1 到 SPOKE2 之间的私网

```
R6#ping 7.7.7.7 source 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.7, timeout is 2 seconds:
Packet sent with a source address of 6.6.6.6
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/5/8 ms
R6#
```

再测试 SPOKE1 背后的私网能否访问 internet

```
R6#ping 4.4.4.4 source 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 6.6.6.6
.....
Success rate is 0 percent (0/5)
```

明显看到不能访问。

细心的朋友会看到，当 SPOKE1 和 SPOKE2 互访时，其私网路由下一跳为 HUB 的 Tunnel 地址，如 SPOKE1 访问 7.7.7.7/24 时，下一跳地址为 100.1.1.5/24，而 SPOKE1 明显可以访问 SPOKE2 的 Tunnel 口地址 100.1.1.6/24，所以，如果将 SPOKE1 访问 7.7.7.7/24 时，下一跳改为 100.1.1.6/24，会更好。那么如何改呢？可以通过修改 OSPF 的类型为 Broadcast

(3)修改 OSPF 类型为 Broadcast

①HUB 修改

```
R5(config)#interface tunnel 567
R5(config-if)#ip ospf network broadcast
R5(config-if)#no shutdown
```

②SPOKE1 修改

```
R6(config)#interface tunnel 567
R6(config-if)#ip ospf network broadcast
R6(config-if)#no shutdown
```

③SPOKE2 修改

```
R7(config)#interface tunnel 567
R7(config-if)#ip ospf network broadcast
R7(config-if)#no shutdown
```

再查看 SPOKE1 和 SPOKE2 上的路由表


```
5.0.0.0/24 is subnetted, 1 subnets
O 5.5.5.0 [110/1001] via 100.1.1.5, 00:24:44, Tunnel567
O 7.0.0.0/24 is subnetted, 1 subnets
O 7.7.7.0 [110/1001] via 100.1.1.7, 00:24:44, Tunnel567
R6#
```

```
5.0.0.0/24 is subnetted, 1 subnets
O 5.5.5.0 [110/1001] via 100.1.1.5, 00:25:17, Tunnel567
O 6.0.0.0/24 is subnetted, 1 subnets
O 6.6.6.0 [110/1001] via 100.1.1.6, 00:25:07, Tunnel567
R7#
```

可以看到此时 SPOKE1 和 SPOKE2 之间的私网路由，都是以对方的 Tunnel 口的 IP 地址为下一跳的，这样，SPOKE1 和 SPOKE2 背后的私网互相访问时，不用再经过 HUB 做中转。

5.配置 NAT, 让 HUB、SPOKE1、SPOKE2 背后的私网访问 Internet

(1)HUB 上配置 NAT

①配置 NAT 抓流量

必须使用 NAT 抓取 HUB 背后的私网访问 Internet 时的流量，而对于 HUB 背后私网与 SPOKE1、SPOKE2 背后私网的流量，丢弃。

```
R5(config)#ip access-list extended 100
R5(config-ext-nacl)#deny ip 5.5.5.0 0.0.0.255 6.6.6.0 0.0.0.255
R5(config-ext-nacl)#deny ip 6.6.6.0 0.0.0.255 5.5.5.0 0.0.0.255
R5(config-ext-nacl)#
R5(config-ext-nacl)#deny ip 5.5.5.0 0.0.0.255 7.7.7.0 0.0.0.255
R5(config-ext-nacl)#deny ip 7.7.7.0 0.0.0.255 5.5.5.0 0.0.0.255
R5(config-ext-nacl)#permit ip any any
R5(config-ext-nacl)#exit
```

②定义 Outside、Inside

```
R5(config)#interface e0/0
R5(config-if)#ip nat outside

R5(config)#interface loopback 0
R5(config-if)#ip nat inside
```

③定义转换

```
R5(config)#ip nat inside source list 100 interface e0/0 overload
R5(config)#
```

测试，HUB 背后的私网能否访问 Internet

```
R5#ping 4.4.4.4 source 5.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/5 ms
R5#
```

可以看到此时访问成功。

(2)SPOKE1 配置 NAT

①配置 NAT 抓流量

```
R6(config)#ip access-list extended 100
R6(config-ext-nacl)#deny ip 6.6.6.0 0.0.0.255 5.5.5.0 0.0.0.255
R6(config-ext-nacl)#deny ip 5.5.5.0 0.0.0.255 6.6.6.0 0.0.0.255
R6(config-ext-nacl)#
R6(config-ext-nacl)#deny ip 6.6.6.0 0.0.0.255 7.7.7.0 0.0.0.255
R6(config-ext-nacl)#deny ip 7.7.7.0 0.0.0.255 6.6.6.0 0.0.0.255
R6(config-ext-nacl)#permit ip any any
R6(config-ext-nacl)#exit
```

②配置 Outside、Inside

```
R6(config)#interface dialer 1
R6(config-if)#ip nat outside
```

注意 SPOKE1 上的 Outside 必须在 dialer1 上配置。

```
R6(config)#interface loopback 0
R6(config-if)#ip nat inside
R6(config-if)#no shutdown
```

③配置转换

```
R6(config)#ip nat inside source list 100 interface dialer 1 overload
R6(config)#
```

同样，做转换时，必须以 dialer 1 进行复用。

```
R6#ping 4.4.4.4 source 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 6.6.6.6
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
R6#
```

(3)SPOKE2 上配置 NAT

①配置 ACL 抓流量

```
R7(config)#ip access-list extended 100
R7(config-ext-nacl)#deny ip 7.7.7.0 0.0.0.255 5.5.5.0 0.0.0.255
R7(config-ext-nacl)#deny ip 5.5.5.0 0.0.0.255 7.7.7.0 0.0.0.255
R7(config-ext-nacl)#
R7(config-ext-nacl)#deny ip 7.7.7.0 0.0.0.255 6.6.6.0 0.0.0.255
R7(config-ext-nacl)#deny ip 6.6.6.0 0.0.0.255 7.7.7.0 0.0.0.255
R7(config-ext-nacl)#permit ip any any
R7(config-ext-nacl)#exit
```

②配置 Outside、Inside

```
R7(config)#interface dialer 1
R7(config-if)#ip nat outside
```

```
R7(config)#interface loopback 0
R7(config-if)#ip nat inside
R7(config-if)#no shutdown
```

③配置转换

```
R7(config)#ip nat inside source list 100 interface dialer 1 overload
R7(config)#
```

```
R7#ping 4.4.4.4 source 7.7.7.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 7.7.7.7
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

可以看到私网可以访问 Internet。

截止现在，HUB、SPOKE1、SPOKE2 背后的私网不仅可以访问 Internet，也可以互相访问。但我们要求私网之间互访时，必须进行数据加密，而私网访问 Internet 时不加密，此时就要在 MGRE 的基础上配置 IPSEC。

6.配置 IPSEC，对数据加密

(1)HUB 配置

```
R5(config)#crypto isakmp enable
R5(config)#crypto isakmp policy 567
R5(config-isakmp)#authentication pre-share
R5(config-isakmp)#encryption aes
R5(config-isakmp)#hash md5
R5(config-isakmp)#group 2
R5(config-isakmp)#exit
R5(config)#crypto isakmp key cisco address 0.0.0.0
```

```
R5(config)#crypto ipsec transform-set 567 esp-aes esp-md5-hmac
R5(cfg-crypto-trans)#mode transport
R5(cfg-crypto-trans)#exit
R5(config)#crypto ipsec profile DMVPN
R5(ipsec-profile)#set transform-set 567
R5(ipsec-profile)#exit
```

```
R5(config)#interface tunnel 567
R5(config-if)#tunnel protection ipsec profile DMVPN
```

(2)SPOKE1 配置

```
R6(config)#crypto isakmp enable
R6(config)#crypto isakmp policy 567
R6(config-isakmp)#authentication pre-share
R6(config-isakmp)#encryption aes
R6(config-isakmp)#hash md5
R6(config-isakmp)#group 2
R6(config-isakmp)#exit
R6(config)#crypto isakmp key cisco address 0.0.0.0
```

```
R6(config)#crypto ipsec transform-set 567 esp-aes esp-md5-hmac
R6(cfg-crypto-trans)#mode transport
R6(cfg-crypto-trans)#exit
R6(config)#crypto ipsec profile DMVPN
R6(ipsec-profile)#set transform-set 567
R6(ipsec-profile)#exit
```

```
R6(config)#interface tunnel 567
R6(config-if)#tunnel protection ipsec profile DMVPN
```

(3)SPOKE2 配置

```
R7(config)#crypto isakmp enable
R7(config)#crypto isakmp policy 567
R7(config-isakmp)#authentication pre-share
R7(config-isakmp)#encryption aes
R7(config-isakmp)#hash md5
R7(config-isakmp)#group 2
R7(config-isakmp)#exit
R7(config)#crypto isakmp key cisco address 0.0.0.0
```

```
R7(config)#crypto ipsec transform-set 567 esp-aes esp-md5-hmac
R7(cfg-crypto-trans)#mode transport
R7(cfg-crypto-trans)#exit
R7(config)#crypto ipsec profile DMVPN
R7(ipsec-profile)#set transform-set 567
R7(ipsec-profile)#exit
```

```
R7(config)#interface tunnel 567
R7(config-if)#tunnel protection ipsec profile DMVPN
R7(config-if)#no shutdown
```

此时，在 HUB、SPOKE1、SPOKE2 上查看，IPSEC 是否配置起来

```
R5#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state        conn-id  status
36.1.1.5    25.1.1.5    QM_IDLE     1002    ACTIVE
25.1.1.5    36.1.1.5    QM_IDLE     1001    ACTIVE
25.1.1.5    17.1.1.10   QM_IDLE     1003    ACTIVE
17.1.1.10   25.1.1.5    QM_IDLE     1004    ACTIVE
```

可以看到 HUB 上建立了两条双向的 SA。

```
R6#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state         conn-id status
17.1.1.10   36.1.1.5     QM_IDLE      1004 ACTIVE
36.1.1.5    17.1.1.10   QM_IDLE      1003 ACTIVE
36.1.1.5    25.1.1.5    QM_IDLE      1002 ACTIVE
25.1.1.5    36.1.1.5    QM_IDLE      1001 ACTIVE
```

SPOKE1 上同样。

```
R7#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state         conn-id status
17.1.1.10   36.1.1.5     QM_IDLE      1004 ACTIVE
36.1.1.5    17.1.1.10   QM_IDLE      1001 ACTIVE
17.1.1.10   25.1.1.5    QM_IDLE      1003 ACTIVE
25.1.1.5    17.1.1.10   QM_IDLE      1002 ACTIVE
```

此时，说明 IPSEC 建立成功。

QQ:1786046994